

---

This is the **published version** of the article:

Giudice, Eugenia Lo; Galofré Pujol, Maria; Piñero, Moisés; [et al.]. Aprendiendo a localizar : OmegaT e Ignuit. 2019. 83 p.

---

This version is available at <https://ddd.uab.cat/record/203605>

under the terms of the  license

# Aprendiendo a localizar: OmegaT e Ignuit

Traducción de Productos Digitales



**Eugenia Lo Giudice**

<MariaEugenia.Giudice@e-campus.uab.cat>

**Mariona Galofré**

<Maria.Galofre@e-campus.uab.cat>

**Moisés Piñero**

<Moises.Pinero@e-campus.uab.cat>

**Tatiana Galli**

<TatianaAndrea.Galli@e-campus.uab.cat>

Prof.: Adrià Martín-Mor

Máster Tradumática – Tecnologías de la Traducción

2018 – 2019

Universidad Autónoma de Barcelona

## Índice de contenidos

Índice de contenidos .....	2
Índice de ilustraciones.....	4
Tabla de siglas .....	7
1. Introducción .....	1
1.1. Cuadro resumen .....	2
1.2. <i>Software</i> libre y <i>software</i> de código abierto: marco teórico .....	3
1.2.1. Software libre .....	4
1.2.2. <i>Software</i> de código abierto .....	4
1.3. Presentación del grupo .....	5
1.4. Presentación de los programas escogidos .....	6
1.4.1. OmegaT 4.1.5 (2) .....	6
1.4.2. Iguit 2.24.3 .....	7
1.5. Proceso de elección y justificación de los programas escogidos .....	9
1.5.1. Criterios de selección .....	12
1.5.2. El porqué de nuestra elección final .....	13
1.6. Elección de herramientas .....	14
1.6.1. Otras herramientas TAO valoradas .....	16
1.6.2. Herramientas complementarias .....	18
2. Gestión del proyecto .....	19
2.1. Planteamiento del proyecto .....	19
2.1.1. Uso de la plataforma GitHub .....	19
2.2. Roles y reparto del trabajo .....	20
2.3. Flujo de trabajo .....	21
2.3.1. Primeros pasos del proyecto OmegaT .....	22
2.3.2. Dos fases diferentes para el proyecto Iguit .....	24
3. Fase de preparación .....	25
3.1. Introducción al análisis del TO .....	25
3.2. Análisis lingüístico de los originales .....	26
3.3. Análisis técnico de los originales .....	27
3.3.1. Identificación de los archivos traducibles .....	28
3.3.2. Recuento de palabras .....	35
3.4. Preparación de materiales de referencia .....	39

3.5. Selección de guías de estilo.....	42
3.6. Instalación de un sistema operativo en un <i>Live USB</i> y de Ignuit.....	43
3.7. Configuración de preferencias en OmegaT.....	43
4. Fase de traducción .....	45
4.1. Aspectos lingüísticos .....	45
4.2. Aspectos de formato y localización.....	50
5. Fase de postproducción .....	53
5.1. Revisión .....	53
5.1.1. Revisión individual.....	53
5.1.2. Revisiones cruzadas.....	57
5.1.3. Revisión conjunta .....	57
5.2. Comprobación técnica .....	58
5.3. Testeo.....	58
5.3.1. OmegaT .....	58
5.3.2. Ignuit .....	60
6. Generación de los materiales finales y entrega .....	62
7. Conclusiones.....	67
8. Bibliografía .....	71

## Índice de ilustraciones

Ilustración 1. Interfaz de Ignuit. Ventana «Quiz».....	8
Ilustración 2. Interfaz de Ignuit. Ventana «Find» .....	8
Ilustración 3. Contenido del archivo «README» de Ignuit .....	9
Ilustración 4. Contenido del archivo «README» de Ignuit (2).....	9
Ilustración 5. Ejemplos de <i>locales</i> de WordPress.....	11
Ilustración 6. Interfaz de OmegaT parcialmente traducida al catalán .....	13
Ilustración 7. Interfaz de Crowdin. Carpetas de código de Wire iOS .....	14
Ilustración 8. Interfaz de Crowdin. Algunas lenguas de localización disponibles .....	15
Ilustración 9. Interfaz de Crowdin. Cadenas de texto de Wire iOS .....	15
Ilustración 10. Interfaz de MemoQ. Archivo de interfaz de Ignuit .....	16
Ilustración 11. Interfaz de Lokalise. Archivo de interfaz de Ignuit.....	17
Ilustración 12. Interfaz de PoEdit. Archivo de interfaz de Ignuit .....	17
Ilustración 13. Interfaz de GTranslate. Archivo de interfaz de Ignuit .....	17
Ilustración 14. Proyectos de localización de OmegaT en GitHub .....	20
Ilustración 15. Creación de un repositorio en GitHub .....	20
Ilustración 16. Interfaz de OmegaT. Descarga de proyecto en equipo.....	23
Ilustración 17. Conexión de OmegaT con un repositorio en línea .....	23
Ilustración 18. Interfaz de OmegaT. Abrir proyecto reciente .....	23
Ilustración 19. Interfaz de OmegaT. Ventana de preferencias. ID del traductor .....	24
Ilustración 20. Sitio web de OmegaT. Archivos traducibles de OmegaT .....	28
Ilustración 21. Sitio web de OmegaT. Instrucciones para localizadores .....	28
Ilustración 22. «Full package» en Source Forge de la localización de OmegaT.....	29
Ilustración 23. Interfaz de OmegaT. Ventana de archivos traducibles .....	29
Ilustración 24. Carpetas de archivos de OmegaT.....	30
Ilustración 25. Archivos de la subcarpeta «source».....	30
Ilustración 26. Estructura de un archivo de OmegaT con extensión «.properties».....	30
Ilustración 27. Dificultad de identificación de texto traducible en «Bundle.properties» .....	31
Ilustración 28. Interfaz de OmegaT. Archivos reconocidos como traducibles.....	32
Ilustración 29. Archivos en la carpeta «po» .....	33
Ilustración 30. Vista del archivo «en_US.po» de Ignuit en Notepad++ .....	33
Ilustración 31. Vista del archivo «LINGUAS» de Ignuit en Notepad++.....	34
Ilustración 32. Archivos traducidos de Ignuit.....	34
Ilustración 33. Interpretación y visualización de las etiquetas de formato en OmegaT.....	35
Ilustración 34. Interfaz de OmegaT. Función «Statistics» .....	37
Ilustración 35. Interfaz de WinMerge. Comparación de estadísticas .....	37
Ilustración 36. Interfaz de MemoQ. Estadísticas de «en_US.po», «ignuit.xml» y «README.txt» .....	38
Ilustración 37. Interfaz de Matecat. Estadísticas de «en_US.po», «ignuit.xml» y «README.txt» .....	38
Ilustración 38. Interfaz de OmegaT. Estadísticas de «en_US.po», «ignuit.xml» y «README.txt» .....	38
Ilustración 39. Interfaz de OmegaT. Estadísticas del proyecto de OmegaT (EN>CA) con archivo POT propagado.....	39

Ilustración 40. Interfaz de OmegaT. Estadísticas del proyecto de OmegaT (EN>IT) con archivo POT propagado.....	39
Ilustración 41. Memorias de traducción para el proyecto de OmegaT en la carpeta «auto» (EN>CA) .....	39
Ilustración 42. Memorias de traducción para el proyecto de OmegaT en la carpeta «tm» (EN>ES) .....	40
Ilustración 43. Extracción de términos de Ignuit con Five Filters .....	42
Ilustración 44. Extracción de términos de Ignuit con Five Filters (2) .....	42
Ilustración 45. Extracción de términos de Ignuit con Vocab Grabber .....	42
Ilustración 46. Creación de un Live USB con Linux Live USB Creator .....	43
Ilustración 47. «Allow translation to be equal to source» .....	44
Ilustración 48. Casilla de verificación para la función de autopropagación en las preferencias de OmegaT .....	44
Ilustración 49. Preferencias en OmegaT. Comportamiento frente a segmentos no traducidos: insertar el texto original o dejar el segmento meta vacío .....	44
Ilustración 50. Función « <i>auto-text</i> » en las preferencias de OmegaT .....	45
Ilustración 51. Preferencias de OmegaT. Habilitación del texto predictivo.....	45
Ilustración 52. Ejemplo del cambio de sujeto en la oración (OmegaT, EN>CA).....	46
Ilustración 53. Ejemplo de despersonificación en la traducción EN>IT de OmegaT.....	46
Ilustración 54. Traducción al ES de « <i>No cards in Trash</i> » (Ignuit) .....	46
Ilustración 55. Inclusión de artículo en la traducción EN>CA de OmegaT .....	46
Ilustración 56. Inclusión de artículo en la traducción EN>IT de OmegaT.....	46
Ilustración 57. Inclusión de artículo en la traducción EN>CA de OmegaT .....	47
Ilustración 58. Vista del uso de la función de búsqueda externa en OmegaT .....	47
Ilustración 59. Omisión de signo de exclamación en la traducción EN>CA de OmegaT .....	47
Ilustración 60. Omisión de signo de exclamación en la traducción EN>IT de Ignuit.....	47
Ilustración 61. Cambio en la estructura de la oración en la traducción EN>CA de OmegaT .....	47
Ilustración 62. Elisión de signos de puntuación excesivos en la traducción EN>CA de OmegaT .....	48
Ilustración 63. Traducción vacía de « <i>plase</i> » en la traducción EN>CA de OmegaT .....	48
Ilustración 64. Traducción vacía de « <i>plase</i> » en la traducción EN>ES de Ignuit .....	48
Ilustración 65. Traducción vacía de « <i>plase</i> » en la traducción EN>IT de Ignuit .....	48
Ilustración 66. Uso de determinante demostrativo en la traducción EN>CA de OmegaT.....	48
Ilustración 67. Uso de artículo definido en la traducción EN>CA de OmegaT .....	48
Ilustración 68. Uso de artículo definido en la traducción EN>CA de OmegaT .....	48
Ilustración 69. Diferencias conceptuales EN>CA extraídas de la traducción EN>CA de OmegaT.....	49
Ilustración 70. Uso de mayúsculas y minúsculas en la traducción EN>ES de Ignuit .....	49
Ilustración 71. Uso de mayúsculas y minúsculas en la traducción EN>IT de Ignuit .....	49
Ilustración 72. Transposición en la traducción EN>IT de OmegaT.....	49
Ilustración 73. Acentos diacríticos. Falso positivo en el QA con OmegaT: «ves», del verbo «ir» en catalán .....	49
Ilustración 74. Preferencias de corrección ortográfica del LanguageTool de OmegaT: dejar de corregir el diacrítico en «ves» en catalán .....	50
Ilustración 75. Término « <i>sticky flips</i> » sin traducir y segmento « <i>Tag</i> » interpretado como verbo en la traducción EN>ES de Ignuit .....	51
Ilustración 76. Segmento « <i>Listen to Front Automatically</i> » de Ignuit traducido al IT.....	51
Ilustración 77. Presencia de <i>hotkeys</i> en la traducción de OmegaT.....	52
Ilustración 78. <i>Hotkeys</i> y marcador «gtk» en segmentos de Ignuit.....	52
Ilustración 79. Ejemplo de cadena con marcador «gtk» en la traducción EN>IT de Ignuit .....	52

Ilustración 80. Referencia a un archivo de la documentación de OmegaT en cadenas de la interfaz de usuario (EN>ES).....	52
Ilustración 81. Referencia a un archivo de la documentación de Ignuit en cadenas del archivo «README.txt» (EN>ES) .....	52
Ilustración 82. Número total de errores en el primer QA de la traducción EN>ES de OmegaT .	53
Ilustración 83. Número de errores en el primer QA del archivo «en_US.po» en la traducción EN>IT de Ignuit .....	53
Ilustración 84. Número total de errores en la traducción EN>ES de OmegaT .....	54
Ilustración 85. Archivo «project_save.tmx» de la traducción EN>ES de OmegaT en el repositorio GitHub .....	54
Ilustración 86. Muestra de una consulta al «project_save.tmx» en GitHub de la traducción EN>ES de OmegaT .....	55
Ilustración 87. Segmento « <i>S&amp;pace matches nbsp</i> » en la traducción EN>CA de OmegaT .....	55
Ilustración 88. Búsqueda con Ctrl+F de la expresión «nbsp» en el «project_save.tmx» de la traducción EN>ES de OmegaT en GitHub .....	55
Ilustración 89. Dudas sobre la traducción de Ignuit en el archivo compartido para todas las lenguas .....	56
Ilustración 90. Dudas sobre la traducción de Ignuit en el archivo compartido para todas las lenguas (2) .....	56
Ilustración 91. Dudas sobre la traducción de Ignuit en el archivo compartido para todas las lenguas (3) .....	56
Ilustración 92. Captura del documento compartido en el que se vertieron las dudas para un posterior correo para el desarrollador.....	57
Ilustración 93. Archivo «it.pot» (traducción al IT de «ignuit.pot») en Notepad++ .....	58
Ilustración 94. Terminal del sistema operativo Tradumàtix. Aplicación de nuestras traducciones a la GUI de OmegaT.....	59
Ilustración 95. Fase de testeo. Interfaz de OmegaT traducida al catalán .....	59
Ilustración 96. Fase de testeo. Interfaz de OmegaT traducida al catalán (2).....	59
Ilustración 97. Captura de un correo del desarrollador de Ignuit con instrucciones sobre el testeo .....	60
Ilustración 98. Fase de testeo. Nuestra traducción del menú « <i>Vista</i> » (« <i>View</i> ») en la GUI de Ignuit en IT.....	60
Ilustración 99. Fase de testeo. Interfaz de Ignuit traducida al español .....	60
Ilustración 100. Fase de testeo. Interfaz de Ignuit traducida al catalán .....	60
Ilustración 101. Fase de testeo. Interfaz de Ignuit traducida al catalán (2).....	61
Ilustración 102. Archivos de imagen de la carpeta «icons» del paquete de descarga de Ignuit	61
Ilustración 103. <i>Hotkeys</i> en la GUI de Ignuit en IT .....	62
Ilustración 104. Segmento « <i>Text colours</i> » en OmegaT.....	62
Ilustración 105. Comprobación de la funcionalidad de la etiqueta para texto en negrita en la GUI de Ignuit .....	62
Ilustración 108. Descarga de Olifant desde Source Forge .....	63
Ilustración 109. Asistente de instalación de Olifant. Instalación completada .....	63
Ilustración 110. Menú « <i>File</i> » en Olifant para abrir nuestra MT .....	64
Ilustración 111. Interfaz de Olifant con nuestra MT del proyecto EN>ES de OmegaT abierta...	64
Ilustración 112. Menú « <i>View</i> » en Olifant. Selección de la función « <i>Fields Settings</i> ».....	64
Ilustración 113. Elección de los metadatos de la MT que queremos que muestre Olifant .....	65
Ilustración 114. Vista de la MT del proyecto EN>ES de OmegaT con los metadatos seleccionados .....	65

Ilustración 115. Botón « <i>Filter Settings</i> » de Olifant .....	65
Ilustración 116. Interfaz de Olifant. Filtro para seleccionar autoraje del último cambio en la MT .....	66
Ilustración 117. Fórmula «ChgUser_ES = `XYZ'» en Olifant .....	66
Ilustración 118. Selección de múltiples ID de usuario para la función «ChgUser_ES = `XYZ'» en Olifant.....	66
Ilustración 119. MT del proyecto EN>ES de OmegaT únicamente con segmentos modificados por las ID de usuario «eumeu» y «taty» (Eugenia y Tatiana) .....	67
Ilustración 120. Opciones de exportación de la MT desde el asistente de Olifant.....	67

## Tabla de siglas

Siglas	Término completo
<i>API</i>	<i>application programming interface</i>
<i>CMS</i>	<i>content management system</i>
<i>FOSS</i>	<i>free open source software</i>
<i>GNU GPL</i>	<i>GNU General Public License</i>
<i>GNU FDL (o GFDL)</i>	<i>Gnu Free Documentation License</i>
<i>GUI</i>	<i>graphic user interface</i>
MT	memoria de traducción
<i>NMT</i>	<i>neural machine translation</i>
PO	(extensión de archivo): objeto portable ( <i>portable object</i> )
POT	(extensión de archivo): plantilla para objeto portable ( <i>portable object template</i> )
<i>QA</i>	<i>quality assurance</i>
SGET	sistema de gestión y edición de traducciones
SO	sistema operativo (OS en inglés)
TA	traducción automática
TAE	traducción automática estadística
TABR	traducción automática basada en reglas
TAO	traducción asistida por ordenador
TMX	(extensión de archivo, del inglés) <i>translation memory exchange</i>



TO	texto original
TPD	traducción de productos digitales
UAB	Universitat Autònoma de Barcelona (Universidad Autónoma de Barcelona)
VCS	<i>version control system</i>

Con especial agradecimiento a Marc Riera Irigoyen, compañero de clase y consejero, por acceder a asesorarnos a lo largo del desarrollo de este trabajo.

Nota de los autores: Las alusiones al desarrollador de Ignuit del presente escrito han sido incluidas con su conocimiento y bajo su consentimiento.

## 1. Introducción

Las tecnologías están a la orden del día y han supuesto una revolución en la industria de la traducción. Dentro de este marco, podemos situar la localización de programas informáticos.

En la presente memoria, explicamos el proceso de localización parcial de dos programas de *software* libre: la herramienta de traducción asistida [OmegaT](#) y una aplicación para crear tarjetas mnemotécnicas llamada [Ignuit](#) desarrollada en base al sistema operativo GNOME 2 (entorno de escritorio de Gnu/Linux). Las traducciones se realizaron en ambos casos partiendo del inglés. Las lenguas meta fueron el español y el catalán para OmegaT y el español, el catalán y el italiano para Ignuit.

Nuestro grupo está compuesto por: Moisés Piñero, Tatiana Galli, Eugenia Lo Giudice y Mariona Galofré, todos titulados en Traducción. El equipo cubre varias lenguas de trabajo, siendo el inglés y el español comunes a todos los integrantes. Cada uno de nosotros ha podido aportar valores distintivos al proyecto.

Por motivos que explicamos más extensamente en el apartado [1.5](#), acabamos escogiendo dos programas en vez de uno. En el caso de OmegaT, quisimos aportar nuestro grano de arena a una localización ya en curso; en el caso de Ignuit, la razón de la elección es que nos había parecido interesante el programa en sí. Por el volumen de trabajo que suponían las dos traducciones juntas, nuestra contribución no pudo culminar en una localización total de ninguna de las dos herramientas, aunque intentamos traducir el máximo de cadenas de texto posible.

Ciertamente, podríamos haber dedicado una memoria escrita separada para cada uno de los dos programas, puesto que, en realidad, podemos considerar que hemos desarrollado dos proyectos independientes. Aun así, hemos aprovechado recursos de un proyecto para el otro y viceversa; el método de trabajo ha sido el mismo y el grupo ha intentado siempre «sincronizar» las dos traducciones a fin de conseguir una especie de fusión entre las dos tareas, siempre buscando la consistencia interna como equipo.

Ninguno de los integrantes tenía experiencia previa en el mundo de la localización de programas informáticos, por lo que el trabajo ha resultado ser un enorme reto con algunas dificultades, si bien a la vez ha sido una aventura enriquecedora y muy didáctica. Antes de empezar, como es natural, se nos había puesto en contexto, se nos había explicado un marco teórico y habíamos recibido una serie de instrucciones; a partir de ahí, se nos concedía el «libre albedrío», incluyendo la elección de los programas a localizar. Este enfoque nos ha brindado la oportunidad de adentrarnos de manera totalmente práctica en las especificidades de un campo profesional que era nuevo para nosotros. Para resolver nuestras dudas o avanzar en el proceso, hemos tenido que investigar, probar —y fallar en muchos casos—, preguntar, observar...; esto es lo que nos ha permitido aprender y adquirir destrezas que antes no teníamos. Por otro lado, debemos recalcar que no todos los conocimientos que hemos aplicado los hemos aprendido de cero; también hemos podido emplear nociones de otras asignaturas cursadas hasta el momento en relación con diferentes ámbitos (bases de datos terminológicas, sistemas de gestión y edición de traducciones, herramientas de traducción asistida, ingeniería de la localización, traducción automática, etc.).

Lo que pretendimos al plantear nuestro trabajo fue simular, en la medida de lo posible, un encargo real de traducción de *software* —aunque dejando de lado las cuestiones de tarificación y facturación—. Nos basamos en las fases que hemos estudiado en el Máster Tradumática, desde la recepción del encargo —elección de programas a traducir, en este caso— hasta la generación de los archivos en lengua meta y la preparación de todos los materiales a entregar. Somos conscientes, de todas formas, de que por nuestra falta de experiencia algunos procedimientos quizá no se han llevado a cabo de la forma más eficiente o del mismo modo en que lo harían individuos o agencias con más bagaje.

Mediante esta tarea, hemos descubierto el universo de los proyectos colaborativos. Las traducciones que realizamos constituyen una aportación voluntaria al código de los programas mencionados.

Los materiales traducidos quedan especificados en el apartado [3.3.1](#) del presente trabajo. Incluyen tanto cadenas de las interfaces de usuario como archivos de documentación.

En las páginas que siguen, explicamos cómo se llevó a cabo todo el proceso, desde la decisión de qué íbamos a traducir y con qué medios hasta la producción de los archivos traducidos y el testeo de nuestras traducciones. Consideramos este trabajo nuestro debut en el sector de las tecnologías de la traducción. Lo que podría verse como un mero ejercicio de clase es en realidad una tarea con repercusión en el «mundo real», porque nuestras traducciones podrán ser aplicadas a los programas y publicadas —a falta todavía de algunos retoques—. Esta memoria pretende ser un testimonio escrito de nuestra experiencia, más o menos a modo de crónica —de ahí el título *Aprendiendo a localizar*—, pero con tono y voluntad académicos.

Esperamos que nuestro trabajo pueda ser de ayuda a otros estudiantes o a cualquier persona interesada en la localización de programas.

### 1.1. Cuadro resumen

Nombre del programa	OmegaT	Ignuit
<b>Miembros</b>	Moisés Piñero, Tatiana Galli, Eugenia Lo Giudice, Mariona Galofré	
<b>Combinación lingüística</b>	EN>ES EN>CA (No traducimos al IT, puesto que no había demanda)	EN>ES EN>CA EN>IT
<b>Materiales traducidos</b>	ES: archivos de documentación CA: interfaz de usuario + archivos de documentación (En las dos lenguas, completamos traducciones preexistentes)	Interfaz de usuario + algunos archivos de documentación (no existía traducción previa a nuestras lenguas; partíamos de cero)
<b>N.º de palabras del original</b>	75.097 (EN>CA) / 74.985 (EN>ES)*	(Recuento de OmegaT) 5.693 (5.120 sin repeticiones)
<b>N.º de palabras traducidas</b>	ES: 2.928 CA: 4.626	ES: 2.223 CA: 4.073 IT: 2.129
<b>Roles</b>	<ul style="list-style-type: none"> <li>● Moisés: ingeniería, gestión terminológica, gestión de memorias de traducción, evaluación de recursos, análisis de formatos, traducción (EN&gt;CA), revisión y control de calidad, verificación técnica</li> <li>● Eugenia: gestión terminológica, traducción (EN&gt;ES), revisión y control de calidad, evaluación de recursos, verificación técnica, comunicación con los desarrolladores</li> <li>● Mariona: gestión del proyecto, ingeniería, preparación de originales, análisis de datos, análisis de formatos, gestión de memorias de traducción, evaluación de recursos, traducción (EN&gt;CA), revisión y control de calidad, verificación técnica, comunicación con los desarrolladores</li> <li>● Tatiana: gestión terminológica, traducción (EN&gt;IT), revisión y control de calidad, verificación técnica, evaluación de recursos</li> </ul>	

<b>Herramienta de traducción</b>	OmegaT	OmegaT
<b>Otros programas o recursos importantes</b>	<ul style="list-style-type: none"> <li>• MemoQ, Matecat, Notepad++, Okapi Olifant, WinMerge</li> </ul>	<ul style="list-style-type: none"> <li>• MemoQ, Memsource, MateCAT, Notepad++, Lokalise, Poedit, GTranslate</li> <li>• AntConc</li> <li>• Programas de hojas de cálculo</li> </ul>
	<ul style="list-style-type: none"> <li>• Telegram, Google Drive, Yahoo Mail, OneDrive, Skype</li> </ul>	
<b>Sistemas operativos</b>	Windows, Tradumàtix	Windows, Ubuntu 16.04 Xenial, Tradumàtix
<b>Materiales de referencia</b>	<ul style="list-style-type: none"> <li>• Materiales del repositorio GitHub (memorias de traducción y carpeta del proyecto)</li> <li>• Memorias y glosarios de Softcatalà (solo CA) como materiales externos de referencia (no se añadieron a la carpeta del proyecto compartido en GitHub)</li> </ul>	<ul style="list-style-type: none"> <li>• Memorias de traducción: Android (solo ES), KDE, Ubuntu, GNOME</li> <li>• Glosarios: Microsoft, glosario propio</li> <li>• Diccionario: OpenOffice</li> <li>• Memorias y glosarios de Softcatalà (solo CA) como materiales externos de referencia no se añadieron a la carpeta del proyecto)</li> </ul>
<b>Guías de estilo</b>	ES: GNOME CA: GNOME (Softcatalà)	ES, IT: GNOME CA: GNOME (Softcatalà)

\*El recuento que proporcionó OmegaT no dio un número total de palabras igual para una combinación que para la otra.

## 1.2. Software libre y software de código abierto: marco teórico

Como es bien sabido, en la actualidad las tecnologías están presentes en todos los aspectos de nuestra vida de una manera muy evidente. Lawrence Lessig, autor del prólogo de [Software libre para una sociedad libre](#), de Richard M. Stallman (2004), apunta incluso que «[las] máquinas nos dirigen», y que «el código dirige estas máquinas».

En el contexto del presente trabajo, entendemos como «código» el conjunto de instrucciones que dirigen el funcionamiento de una computadora —un ordenador o «las máquinas», en palabras de L. Lessig—. Este código es inicialmente cifrado en forma de texto y posteriormente decodificado y reinterpretado por el ordenador como una serie de impulsos eléctricos que determinarán su comportamiento. No es nuestra intención extendernos en el desarrollo de esta cuestión, aunque sí creemos oportuno mencionar algunos conceptos.

Detrás de la noción de «software» subyacen muchos otros elementos y matices sobre los cuales tampoco podemos profundizar en la presente memoria. Principalmente, tendremos en cuenta una acepción y algunas características básicas de este término.

Según la [Real Academia Española](#), el *software* se define como el «conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora». Ampliando un poco más esta definición con información de la [Wikipedia](#), «el *software* es toda la información procesada por los sistemas informáticos: programas y datos, [y] abarca todo lo intangible, todo lo “no físico”», en contraposición al «hardware», o componentes físicos de la computadora.

Cuando nos referimos a *software*, hablamos de «[código fuente](#)» para hacer referencia a [líneas de texto](#) que dan instrucciones al ordenador sobre acciones que debe ejecutar. Estas líneas de texto no son más que caracteres que se pueden escribir por medio de diferentes

herramientas. La más básica de ellas sería un editor de texto. Otra pieza cabal del concepto de «código fuente» es el «[lenguaje de programación](#)». Simplificando, serían las diferentes estructuras que pueden seguir las líneas de texto. Existen múltiples ejemplos de lenguajes de programación. Entre los más habituales encontramos Java, Python, C++ o JavaScript.

Podemos leer en la Wikipedia que son los programadores informáticos los encargados de escribir el código fuente. Según nuestra experiencia, haría falta puntualizar que hoy en día no es necesariamente así: en tiempos recientes un gran número de personas posee suficientes conocimientos en informática como para programar o editar código fuente. Esto es especialmente cierto en el campo de lo que se conoce como «*software* libre» y en el del «*software* de código abierto».

### 1.2.1. Software libre

El principio básico del *software* libre (*free software*) es que su código fuente puede ser consultado por todos; no solo eso, sino que puede a la vez ser modificado por todos. Todo programa es obra de una persona. Como tal, está sujeto a determinadas licencias. Los productos de *software* libre se elaboran y divulgan bajo licencias de carácter público, en contraste con las licencias privadas y, obviamente, más restrictivas, del «*software* privativo».

El programador y activista estadounidense [Richard Matthew Stallman](#) ha presentado en numerosas obras, conferencias, eventos y redes los fundamentos de la filosofía del *software* libre: «Los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar el *software* y distribuirlo modificado. [...] significa, entre otras cosas, que no tiene que pedir o pagar el permiso. Más precisamente, significa que los usuarios de programas tienen las cuatro libertades esenciales». ¿Cuáles serían, entonces, esas cuatro libertades del *software* libre o los programas libres? Veámoslo a continuación:

- Libertad 0: ejecutar el programa.
- Libertad 1: estudiar cómo trabaja el programa y cambiarlo para que haga lo que el usuario quiera.
- Libertad 2: redistribuir copias del programa.
- Libertad 3: distribuir copias de las versiones modificadas del programa a terceros.

Como bien destaca Stallman (2004)<sup>1</sup>, «libre» no es un sinónimo de gratuito. Todo programa libre puede estar sometido a distribución comercial. El énfasis recae en la posibilidad de acceder al código, de editarlo y de redistribuir copias de las versiones editadas.

### 1.2.2. Software de código abierto

El *software* de código abierto (*open source software*) se interpreta dentro del contexto del *software* libre. Hace referencia concretamente al hecho de que el código fuente sea visible y editable.

Una de las grandes ventajas del código abierto es precisamente esta: cualquier persona, desde cualquier lugar, puede participar en el desarrollo de herramientas informáticas, ya sea propiamente ayudando a desarrollar la base del código o, como en nuestro caso, realizando tareas paralelas e igualmente importantes, como la traducción.

El conjunto de programas libres de código abierto se conoce en inglés como «*free open source software*» (o «*FOSS*», por sus siglas).

Por razones obvias de derechos, para este trabajo se nos asignó la tarea de localizar una herramienta o un producto con licencia libre y de código abierto. Esto también permitió el acceso fácil y cómodo a todos los materiales e información necesarios para la localización.

---

<sup>1</sup> Stallman, Richard M. (2004). *Software libre para una sociedad libre*.

### 1.3. Presentación del grupo

Como hemos dicho, nuestro grupo está compuesto por Moisés, Eugenia, Mariona y Tatiana.

Moisés y Mariona se graduaron en Traducción e Interpretación en la Universidad Autónoma de Barcelona (UAB). Eugenia es traductora jurada nacional de Argentina titulada por la Universidad Nacional de Catamarca. Tatiana cursó la carrera en la Scuola Superiore per Mediatori Linguistici de Ancona (Italia). El hecho de haber estudiado todos Traducción e Interpretación hace que tengamos conocimientos comunes, lo cual ha resultado muy conveniente. Asimismo, no todos estudiamos en la misma universidad ni en la misma promoción, cosa que nos permite sumar lo que se enseña en diferentes instituciones o años académicos. Entre los cuatro, contamos con competencias en tres variantes dialectales de español y dos de catalán, cosa que aporta riqueza a nuestras traducciones y favorece que podamos llegar a un mayor número de usuarios.

Todos somos, como es de imaginar, muy aficionados al estudio de lenguas y al aprendizaje sobre otras culturas. De igual forma, nos une la curiosidad por las nuevas tecnologías y las aplicaciones que estas puedan tener en el mundo de la traducción.

A la vez, cada uno tiene unos elementos particulares —en lo referente a trayectoria, cualidades e inquietudes—, con lo cual todos hemos sumado algún valor al equipo.

En la tabla de abajo se especifican las lenguas de trabajo<sup>2</sup> de cada uno. La marca «(N)» indica el nivel nativo en una lengua concreta.

Moisés	Tatiana	Eugenia	Mariona
Español (N)	Español (N)	Español (N)	Español (N)
Inglés	Inglés	Inglés	Inglés
Catalán (N)	Italiano (N)		Catalán (N)
Alemán	Ruso		Chino

Tabla 1. Lenguas de trabajo de los componentes del grupo.

La lengua de comunicación entre los componentes del equipo ha sido el español.

En un principio, quisimos buscar un programa que nos permitiera aprovechar la variedad de idiomas que entre todos cubríamos. No obstante, esta idea era demasiado ambiciosa, complicada de realizar y poco práctica. Entre otras cosas, habría sido difícil encontrar un programa que cumpliera nuestros criterios y que, además, incluyera todas las lenguas deseadas. Por otro lado, lo más sensato era no escoger ninguna lengua que alguno de los integrantes no conociera en absoluto, porque todos teníamos que ser partícipes de todas las fases del proyecto, y cada uno tenía que poder valorar el trabajo del resto. Otra cuestión evidente y muy relevante es que tal vez podríamos haber traducido desde alguna lengua como el ruso o el chino, pero desde luego no habríamos podido traducir hacia el ruso o el chino, ya que no teníamos destrezas suficientes. Por último, habría sido imposible consensuar términos o revisar traducciones de una lengua de la que la mayoría de integrantes no tenía conocimiento alguno.

En consecuencia, nos centramos en las lenguas comunes, incluyendo el catalán —porque teniendo a dos personas que lo hablaban podían repartirse las responsabilidades de traducción y revisión— y también el italiano, dominado solo por Tatiana —porque la similitud con el catalán y el español al menos nos daba a los demás miembros cierto margen—. Pensamos que si entre todos revisábamos las traducciones al

---

<sup>2</sup> Entendemos por «lenguas de trabajo» aquellas de las que tenemos un dominio suficiente como para traducir a nivel profesional. Hemos dejado aquellas otras que conocemos pero de las que tenemos un nivel inferior.

italiano, analizando en detalle cada segmento con ayuda de Tatiana y valiéndonos de herramientas de control de calidad, podíamos incluir el italiano en nuestro elenco de lenguas sin grandes problemas.

## 1.4. Presentación de los programas escogidos

### 1.4.1. OmegaT 4.1.5 (2)

OmegaT es una herramienta TAO escrita en el lenguaje de programación Java. Como hemos dicho antes, es una herramienta de *software* libre, bajo la Licencia Pública General de GNU ([GNU General Public License](#)) y la Licencia de Documentación Libre de GNU ([GNU Free Documentation License](#)), para la documentación. El código está disponible para ser modificado y mejorado libremente. Su principal creador es Keith Godfrey. Se puede descargar, entre otras fuentes, desde el sitio web oficial. El proyecto de desarrollo de esta herramienta (*OmegaT Project*) está actualmente liderado por Didier Briel.

La localización de OmegaT —interfaz y demás materiales— se ha llevado a cabo, en diferentes niveles de compleción, a 37 lenguas, entre las cuales se encuentran el inglés, el español, el catalán, el italiano, el ruso o el chino (simplificado y tradicional). Todas las versiones en inglés e italiano están muy completas. No es tanto el caso del español y el catalán, y por ello tuvimos la oportunidad de traducir cadenas de texto.

La versión que localizamos es la 4.1.5 (actualización 2). En esos momentos, era la más reciente. Sin embargo, mientras trabajábamos en nuestras traducciones, se lanzó una nueva versión, que es la 4.1.5 (3). Afortunadamente, las traducciones que hicimos se pueden aplicar a esta actualización, así que no representa ningún inconveniente el hecho de que la versión 4.1.5 (2) no fuera la más nueva.

Tanto la localización de esta aplicación como las contribuciones al código en general o los informes sobre errores se efectúan a través de plataformas colaborativas. Para formar parte del equipo de localización, uno debe darse de alta de la lista de distribución ([OmegaT-In10](#)), gestionada mediante Source Forge. Igualmente, cada vez que un colaborador inicia una traducción, puede publicar su intención en el plan de localización ([OmegaT Localization Plan](#)), una lista de distribución de Yahoo Mail.

Usuarios y traductores pueden hacer consultas utilizando el grupo de usuarios ([OmegaT User Group](#), o *OmT*), también de Yahoo Mail.

Más adelante hacemos referencia a otro recurso, la web [How To – Translating OmegaT into your language](#), que es donde se explica lo que hay que hacer para colaborar en la traducción.

OmegaT es un programa con el que hemos trabajado en clase. El Máster Tradumática incluye una asignatura específica para aprender las bases de esta herramienta. Precisamente por esto, nos pareció que participar activamente en el desarrollo y la mejora del programa podía constituir un reto emocionante y gratificante.

OmegaT pertenece al grupo de herramientas conocidas como SGET<sup>3</sup>. Permite crear proyectos nuevos o importar proyectos existentes, así como gestionar y crear memorias de traducción. A continuación, mencionamos algunas de sus características y funciones principales:

- Es una herramienta multiplataforma (funciona en cualquier sistema operativo), es fácil de instalar y utiliza formatos estándar, lo cual hace muy conveniente el intercambio con otras personas y equipos.

- Es un programa sencillo, pero completo, que puede suplir las necesidades de la traducción profesional.

- Tiene una interfaz simple que se puede aprender a usar sin problemas.

---

<sup>3</sup> Término acuñado en: Martin-Mor, A.; Piqué Huerta R.; Sánchez Gijón, P. (2016). *Tradumàtica: tecnologies de la traducció*.



- Busca concordancias desde memorias de traducción, glosarios o el proyecto activo.
- Incorpora texto automático y texto predictivo.
- Desde OmegaT se pueden traducir simultáneamente varios archivos, incluso de diferentes formatos, y consultar en un mismo proyecto diferentes memorias de traducción y glosarios.
- Permite tener activados varios diccionarios.
- Se puede configurar para obtener traducciones de múltiples motores de traducción automática como Apertium, Belazar, DeepL o Google Translate.
- Puede autopropagar traducciones o pretraducir documentos de una manera muy sencilla.
- Se pueden personalizar varias funciones de visualización, como la fuente del texto o los colores de los segmentos traducidos o sin traducir.
- Admite una gran variedad de archivos e incluye un buen conjunto de reglas de filtrado y de segmentación.
- Incorpora la función de alineación de textos.
- Permite crear los documentos traducidos en el formato correspondiente.
- Cuenta con la opción de búsqueda externa, que agiliza el proceso de traducción.
- Se puede conectar a repositorios en línea para que un grupo de personas pueda trabajar simultáneamente en una misma traducción.
- Incluye utilidades para la corrección automática y el control de calidad.
- Permite buscar y reemplazar texto con expresiones regulares.
- Además, incluye:
  - Compatibilidad con Unicode (UTF-8): puede usarse con alfabetos no latinos.
  - Compatibilidad con idiomas con escritura de derecha a izquierda.
  - Compatibilidad con otras aplicaciones y formatos (TMX, TTX, TXML, XLIFF, SDLXLIFF...).

El funcionamiento de OmegaT está explicado en detalle en la página oficial de esta herramienta.

### 1.4.2. Ignuit 2.24.3

Ignuit es un programa de código abierto diseñado para GNU/Linux que permite estudiar vocabulario mediante el uso de tarjetas mnemotécnicas electrónicas —las llamadas [flashcards](#) en inglés—. Concretamente, fue desarrollado para funcionar en GNOME 2. El aspecto de la interfaz se basa en el mismo sistema operativo [GNOME](#).

El desarrollador, Timothy Musson (Tim), es un neozelandés interesado en el aprendizaje de lenguas.

La versión que localizamos es la más reciente: la 2.24.3. Tim Musson ha expresado su voluntad de actualizar el programa y de hacerlo compatible con versiones más nuevas de GNU/Linux.

El programa estaba disponible en inglés (lengua original), ruso y alemán antes del inicio de nuestro proyecto. Nuestro propósito fue sumar el español, el catalán y el italiano de manera íntegra, al menos, en la interfaz de usuario.

Ignuit permite crear tarjetas mnemotécnicas y gestionarlas, agrupándolas en categorías, etiquetándolas, etc. Se pueden adjuntar imágenes, audio, e incluso fórmulas matemáticas a las tarjetas para hacer el proceso de asimilación de contenido más fácil y entretenido.

Estas tarjetas son también llamadas *fichas*, *tarjetas flash*, o, directamente, *flashcards*. En una de las caras, el usuario escribe una o varias palabras en una lengua; en la cara opuesta va la correspondiente traducción. El objetivo es aprender el contenido de las tarjetas para poder dominar mejor el vocabulario de un determinado idioma. El usuario puede examinarse a sí mismo programando pequeños exámenes periódicos o «cuestionarios» —«*quiz*» en la aplicación en inglés—.

El programa está inspirado en un método de memorización y estudio repetitivo conocido como el «[sistema de Leitner](#)». Este sistema fue ideado en la época de la II Guerra Mundial por [Sebastian Leitner](#), psicólogo, militar, periodista y escritor alemán. En sus orígenes, el sistema consistía en escribir una pregunta en la cara frontal de una tarjeta con la respuesta detrás. Las preguntas podían ser de cualquier tema, de manera similar a como funcionan los juegos del estilo de Trivial Pursuit. En el sistema de Leitner, las tarjetas se colocan inicialmente en un primer grupo (la «caja número 1»). Si el usuario logra aprender el contenido de una determinada tarjeta de este grupo, la tarjeta pasa al segundo grupo («caja número 2»). El usuario se va examinando del contenido de las tarjetas de cada grupo o caja. Las tarjetas de la caja número 1 son las que se tienen que estudiar con más frecuencia. El contenido de las cajas posteriores es el que está más asimilado, de manera que los cuestionarios son más espaciados en el tiempo. Si tras un cuestionario no se acierta algo, la tarjeta correspondiente vuelve a la caja 1, con lo cual volverá a aparecer más a menudo en los cuestionarios.

Ignuit sigue esta estructura de una manera moderna e interactiva.

Las siguientes capturas muestran el aspecto de la interfaz<sup>4</sup>.

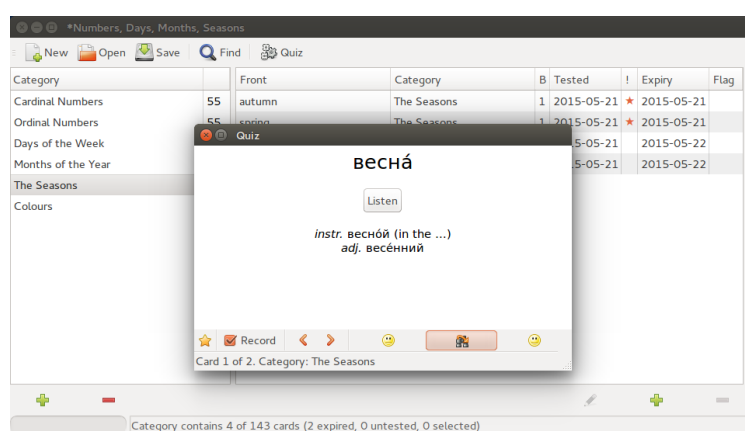


Ilustración 1. Interfaz de Ignuit. Ventana «Quiz»

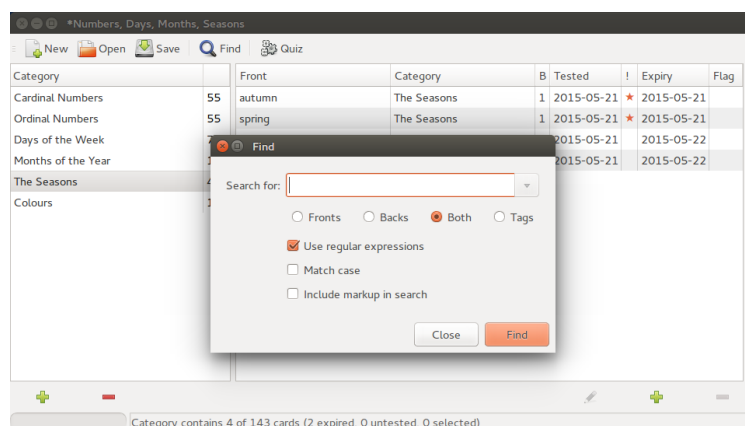


Ilustración 2. Interfaz de Ignuit. Ventana «Find»

Ignuit es una herramienta libre que se puede copiar, distribuir o modificar bajo los términos de la Licencia Pública General de GNU y la Licencia de Documentación Libre de GNU, para la documentación.

Se puede contribuir al código de esta herramienta a través de varias plataformas, tales como [Source Forge](#) o [Savannah](#). Desde estas mismas plataformas se puede descargar todo el material necesario para la instalación y uso del programa.

<sup>4</sup> Imágenes disponibles en el sitio web oficial de Ignuit, proporcionadas por el desarrollador.

Para instalarlo, son necesarios conocimientos básicos de Linux y programación. En las webs de descarga de Ignuit se expone lo que hace falta tener instalado en el ordenador para que Ignuit funcione y se explican las líneas de comandos que hay que escribir en el terminal para poder compilar y ejecutar la aplicación.

```

REQUIREMENTS
=====

This version of ignuit was developed using Ubuntu 16.04. To compile it, you'll
need the development files for:

Library:                                Debian/Ubuntu package name:

libgnomeui-2.0 >= 2.24.5                libgnomeui-dev
libglade-2.0 >= 2.6.4                   libglade2-dev
libxslt >= 1.1.28                       libxslt1-dev
gconf-2.0 >= 3.2.6                      libgconf2-dev
gstreamer >= 1.8.2                      libgstreamer1.0-dev

You'll also need:

gnome-common
gnome-core-devel
intltool (>= 0.50.0)

If you'd like LaTeX support, you'll also need dvipng and LaTeX (e.g. the texlive
package if you're using Debian, Ubuntu, etc). If you're seeing messages like
"[LaTeX render failed]" in Quiz windows, check that you have texlive and dvipng

```

Ilustración 3. Contenido del archivo «README» de Ignuit

```

INSTALLATION
=====

Under GNU/Linux, compilation and installation go something like this:

$ cd ignuit-x.y.z
$ ./configure --prefix=/usr/local --disable-scrollkeeper
$ make
$ sudo make install-strip
$ make clean

If the installation step results in warnings about schemas, try adding the
following arguments to ./configure:

--disable-schemas-install
--disable-scrollkeeper

To uninstall it:

$ cd ignuit-x.y.z
$ sudo make uninstall

```

Ilustración 4. Contenido del archivo «README» de Ignuit (2)

Sin duda, lo más complicado de este proyecto ha sido entender procedimientos de carácter técnico. Los relacionados con Ignuit han sido los más laboriosos: instalar un sistema operativo en un USB, entender cómo compilar e instalar Ignuit en el mismo dispositivo, instalar los componentes necesarios para que Ignuit funcionara, etc. Nos interesaba poder llevar a cabo todas estas operaciones, en primer lugar, para traducir con contexto y, en segundo lugar, para hacer el testeo de las traducciones.

Después de superar una serie de obstáculos, nuestro propósito su cumplió. Explicaremos en otros apartados las acciones que fueron necesarias.

### 1.5. Proceso de elección y justificación de los programas escogidos

La elección de OmegaT e Ignuit no fue nada fácil. La falta de experiencia provocó que en los primeros estadios del proyecto no tuviéramos muchos recursos ni criterios que nos ayudaran a tomar una decisión.

Desde un primer momento tuvimos claro que queríamos localizar un producto que a nosotros mismos, como usuarios, nos interesara. Ello nos permitiría aprender y disfrutar al mismo tiempo. Al final del proceso, también haría posible que usáramos dicho producto con nuestra propia traducción.

Hacíamos frente a la dificultad de escoger un producto digital con licencia libre, cosa que limitaba las posibilidades. Algunos de nuestros planes iniciales tuvieron que ser rechazados por no cumplir esta premisa. Al no saber muy bien por dónde empezar, nos dedicamos sencillamente a realizar una búsqueda genérica en la red, usando palabras o frases clave en diferentes idiomas, como «localización de *software* libre», «*open source software*» y similares. Asimismo, nos apoyamos en ideas extraídas de memorias de trabajo de ediciones anteriores del Máster<sup>5</sup>.

Dentro de este primer paso, accedimos a plataformas colaborativas en línea para indagar entre la inmensa variedad de proyectos publicados en este tipo de sitios web. Hemos de decir que allí encontramos una infinidad de opciones interesantes. De entre ellas, nos interesaba seleccionar preferentemente alguna que tuviera demanda en catalán, español e italiano y que todavía tuviera muchas cadenas de texto pendientes de traducir.

Nombramos a continuación los principales sitios web de trabajo colaborativo con proyectos de *software* libre en los que buscamos:

-[Transifex](#): plataforma de traducción patentada basada en la web. Es un sistema de gestión de la globalización.

-[Savannah](#): entorno web en el que se publican proyectos de *software* libre, principalmente basados en el sistema operativo GNU/Linux. Se pueden hacer aportaciones de diferente naturaleza, tales como contribuciones al código de programas, la redacción de manuales, el testeo o la traducción.

-[GitHub](#): forja para alojar proyectos utilizando el [sistema de control de versiones Git](#). Se utiliza principalmente para la creación de código fuente de programas de ordenador. El *software* que opera GitHub fue escrito en [Ruby on Rails](#).

-[Crowdin](#): plataforma para gestionar proyectos de localización de *software*. Permite buscar proyectos por lenguas de trabajo, por tipología de programa, por tipo de plataforma (ej.: móvil o de escritorio), etc. Además, al consultar las aportaciones que hacen falta para cada programa, muestra el porcentaje de trabajo hecho en cada lengua.

-[PO Editor](#): editor libre, abierto y multiplataforma de catálogos de [Gettext](#) usado en el proceso de localización. Está construido con [wxWidgets](#) y licenciado bajo los términos de la [licencia MIT](#).

-[Softcatalà](#): asociación sin ánimo de lucro con el objetivo de fomentar el uso del catalán en la informática, Internet y las nuevas tecnologías. Nos planteamos buscar a través de esta organización un proyecto en el que hicieran falta otras lenguas, además del catalán, para que todos los integrantes de nuestro equipo pudieran traducir a su lengua nativa. Escribimos un mensaje al correo que figura en la web, pero nunca obtuvimos respuesta.

-[Postgresql](#): sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia [PostgreSQL](#).

-[Source Forge](#): sitio web de colaboración para proyectos de *software*. Fue fundado en 1999 por VA Software y, desde el 18 de septiembre de 2013, es comercializado por Dice Holdings.

Exponemos ahora algunos de los productos que nos planteamos localizar antes de nuestra decisión final:

-[Wikipedia](#): conocida enciclopedia virtual editable por parte de los cibernautas. Pensamos en esta opción por la accesibilidad en cuanto a la edición de este recurso web y porque es un producto muy utilizado).

---

<sup>5</sup> El término «Máster» en el presente documento hace referencia al Máster Tradumática – Tecnologías de la Traducción impartido por la Universidad Autónoma de Barcelona.

-[WordPress](#): sistema de gestión de contenidos o *CMS* (por sus siglas en inglés, *content management system*) enfocado a la creación de cualquier tipo de página web. Igual que con Wikipedia, la idea surgió espontáneamente porque es un producto que todos teníamos bien presente. Wordpress ofrece la opción de colaborar en la edición de sus propios *locales*<sup>6</sup>: en el mismo sitio web podemos encontrar todas las [indicaciones](#) para hacerlo.

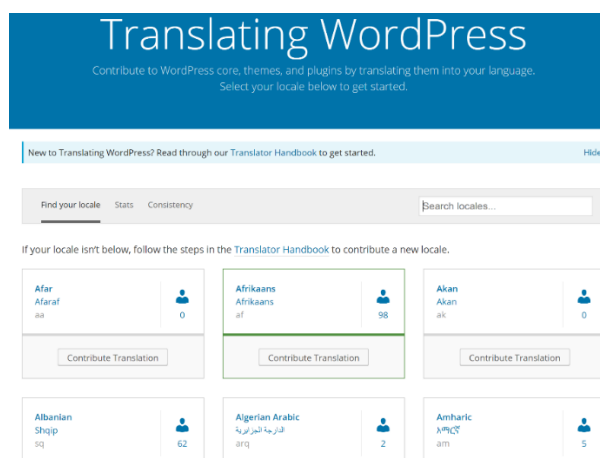


Ilustración 5. Ejemplos de *locales* de WordPress

-[Code](#): organización no gubernamental estadounidense fundada por Hadi y Ali Partovi en 2012. Entre sus objetivos principales se encuentran incluir la informática como materia obligatoria en el plan de educación secundaria estadounidense y la inclusión de las mujeres en las ciencias de la computación.

-[Wire](#): aplicación de mensajería instantánea, intercambio de archivos, llamadas y videollamadas por Internet con cifrado de extremo a extremo<sup>7</sup> por defecto.

-[GDevelo](#)p: *software* gratuito de creación de videojuegos; permite diseñar videojuegos para la web (HTML5) o para Windows y Linux.

-[Taus](#): centro de recursos para el lenguaje global y la industria de la traducción. Valoramos localizar su página web; mas al no recibir respuesta de los desarrolladores, desestimamos la idea.

Paralelamente, tuvimos que llevar a cabo tareas de documentación para lograr discernir cuáles podrían ser mejores opciones que otras. Dos claros ejemplos de ello son los formatos de los originales y las plataformas a través de las cuales se llevaría a cabo la localización. Se hacía indispensable conocer dichos formatos y plataformas para cumplir el objetivo planteado. Poniendo por caso que alguno de los archivos a localizar tuviera la extensión «.php», teníamos que estar mínimamente familiarizados con dicha extensión y escoger una herramienta de traducción que pudiera procesar este tipo de archivo. Del mismo modo, si nos interesaba localizar un producto a través de una plataforma colaborativa, primero teníamos que entender cómo funcionaba la herramienta de edición correspondiente. En este sentido, Wordpress presentaba ciertos inconvenientes, pues ninguno de nosotros había intentado traducir con el editor expresamente diseñado para ello, de manera que algo que debería ser simple e intuitivo se habría convertido en un obstáculo.

<sup>6</sup> El concepto «*locale*» está relacionado con la internacionalización de *software* y hace referencia a la lengua, la cultura o la región para las que se adapta un determinado producto, como podría ser un sitio web.

<sup>7</sup> El cifrado de extremo a extremo (*end-to-end encryption*) en las redes sociales asegura que solo el emisor y el receptor de un mensaje pueden ver dicho mensaje.

### 1.5.1. Criterios de selección

Además de buscar un producto con licencia libre —condición *sine qua non* que debíamos cumplir obligatoriamente—, nos interesaba disponer de contexto. Con esto queremos decir que no solamente teníamos que poder acceder al código del producto, sino que, además, queríamos ver su aspecto al ser utilizado. Por ejemplo, si escogíamos la interfaz de un videojuego, queríamos tener la posibilidad de instalarlo y probarlo antes de traducir dicha interfaz. Los motivos de esto son sencillos: si localizábamos algo con lo que estábamos familiarizados, el proceso se simplificaría y el resultado sería de mejor calidad.

De manera similar, creímos conveniente decantarnos por un producto sobre el cual pudiéramos aplicar nuestra traducción a fin de hacer un *testing* (testeo), cosa que nos permitiría ajustar determinados detalles y pulir la traducción antes de la entrega final.

Ya hemos mencionado que para el grupo era un valor añadido el que el *software* a localizar nos pareciera interesante como usuarios. Nuestras primeras búsquedas en Internet fueron enormemente productivas, pero no todos los materiales que encontramos con demanda de localización nos llamaban la atención como lo hicieron las dos aplicaciones finalmente escogidas.

En otra línea, consideramos muy importante el poder contactar con los desarrolladores y, dependiendo del caso, con otros localizadores. De esta forma, no solo tendríamos más datos útiles para una traducción de calidad; también podríamos hacer consultas en caso de duda, o los mismos desarrolladores nos podrían dar sus propias indicaciones, normas u opiniones.

Con relación a aspectos lingüísticos, en general buscábamos algo que no representara una dificultad excesiva; queríamos priorizar el proceso de aprendizaje en cuanto a herramientas y técnicas para la localización y no tanto la adquisición de terminología o de práctica en la traducción propiamente.

Sabíamos que los aspectos técnicos serían los que representarían más dificultades por nuestra falta de experiencia con las nuevas tecnologías en el mundo de la traducción. Nos tomamos este encargo como la oportunidad perfecta para formarnos más, pero en la elección de programas teníamos que ser cautos y optar por un producto y unas herramientas que pudiéramos llegar a entender y saber utilizar en el plazo de tiempo de que disponíamos.

La necesidad de encontrar un proyecto multilingüe en el que todos tuviéramos la oportunidad de traducir a nuestra lengua nativa —y con suficiente volumen como para cubrir un mínimo de palabras por persona— fue uno de los criterios clave a tener en cuenta. Algunas de las opciones ponderadas al principio requerían traducción solo al español o solo al catalán, por ejemplo. Estas quedaron siempre al final de nuestra lista porque no nos ofrecían a todos la ocasión de traducir a la lengua nativa.

Otra cuestión que nos pareció trascendental es que el producto localizado fuera conocido, ampliamente utilizado, o que pudiera llegar a serlo. Veíamos este punto como una motivación.

En el caso de OmegaT, podemos decir que esta premisa se cumplió, ya que el programa es de uso habitual entre traductores y aficionados a las lenguas; incluso es la herramienta que utilizan algunas empresas de traducción. Sentimos que con nuestra traducción estamos haciendo una contribución valiosa para otros colegas, estudiantes y el público en general.

Quizá esto no se aplica del mismo modo a Iguit, si bien es un programa que podría tener perfecta aceptación entre el público —especialmente si se promocionara y, por encima de todo, si se adaptara a otras versiones de GNU/Linux o incluso a otros sistemas operativos—.

Queremos aprovechar para destacar de nuevo que nuestra colaboración con el desarrollador de Iguit suscitó en él el interés de «reavivar» su propio proyecto. Según nos explicó por correo electrónico, nuestra petición y posterior localización le hicieron plantear actualizaciones del código, mejoras y cambios novedosos en el programa, así como la

posibilidad de aumentar la lista de lenguas para las que Ignuit está disponible —recordemos que, a día de hoy, las lenguas son el inglés, el alemán y el ruso, a falta de acabar de mejorar nuestras traducciones para añadir a la interfaz y a algunos archivos de soporte el español, el catalán y el italiano—.

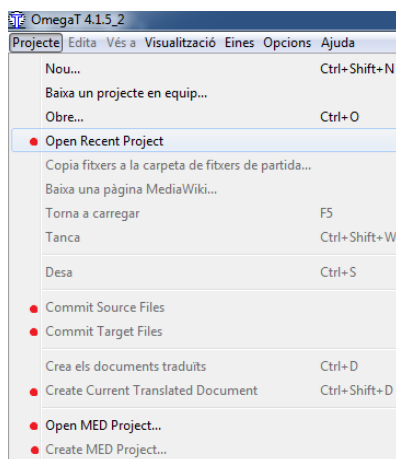
Finalmente —pero no por ello menos importante—, buscábamos un proyecto que nos permitiera poner en práctica las destrezas y conocimientos adquiridos hasta el momento en el Máster.

### 1.5.2. El porqué de nuestra elección final

Durante este proceso de búsqueda e investigación, dimos con Ignuit en la plataforma colaborativa Savannah. Nos pareció interesante localizar un programa que tuviera relación con el estudio de idiomas, ya que nosotros mismos somos apasionados de las lenguas y la traducción y, por decirlo de alguna manera, candidatos perfectos a ser usuarios de Ignuit. Al margen de si podíamos llevar a cabo o no la traducción de Ignuit, la aplicación nos gustó y quisimos descargarla para probarla. En seguida nos dimos cuenta de que instalar Ignuit no era tan fácil como esperábamos. El proceso requería conocimientos de programación básica y uso del terminal de Linux. Antes incluso de llegar a este paso, hacía falta instalar un sistema operativo con el que Ignuit fuera compatible. En el apartado [3.6](#) desarrollamos este punto.

Localizar un programa que ni siquiera podíamos usar nos pareció un gran inconveniente por la falta de contexto, que habría dificultado el proceso y podía afectar negativamente la calidad de los resultados. Consecuentemente, la opción quedó descartada durante cierto tiempo. Más adelante, ante la dificultad de encontrar una herramienta que nos gustara, que viéramos asequible, que requiriera traducción a nuestras lenguas y que a la vez implicara un volumen de trabajo suficiente para los cuatro, decidimos «rescatar» la idea de Ignuit y ponernos en contacto con su desarrollador, Timothy Musson, para contarle sobre el proyecto y preguntarle si estaría dispuesto a ayudarnos. Pensamos que, con algo de suerte, Timothy (Tim) accedería a resolver nuestras dudas sobre aspectos técnicos.

Sin embargo, en ese momento no obtuvimos respuesta de Tim y, viendo las grandes dificultades a las que nos enfrentábamos ya solo para instalar el programa, buscamos otra alternativa y se nos ocurrió la opción de OmegaT. Habíamos usado esta herramienta en las aulas y nos había llamado la atención que la interfaz de usuario estaba disponible en catalán pero traducida parcialmente.



Il·lustración 6. Interfaz de OmegaT parcialmente traducida al catalán

Según descubrimos después, no había necesidades de localización de OmegaT al italiano; en cambio, confirmamos que sí quedaban cosas por hacer en español y catalán. Así pues, decidimos escoger OmegaT como *software* a localizar. Lo que no teníamos claro era si lo que faltaba por localizar bastaría para cubrir nuestras 2.000 palabras por persona.



Fue en esta fase cuando recibimos la respuesta de Tim. Se mostró muy emocionado y totalmente dispuesto a ayudarnos. Pensando que tendríamos poco en lo que trabajar en OmegaT, resolvimos traducir ambos programas. La sorpresa fue cuando descubrimos que, en realidad, en ambos había muchos archivos que traducir, motivo por el que finalmente nuestras traducciones fueron parciales, y no totales.

## 1.6. Elección de herramientas

Decidimos traducir usando OmegaT por varios motivos, los cuales expondremos a continuación. De todas formas, debemos hacer notar que, una vez ya sabíamos que localizaríamos OmegaT y antes de empezar a valorar herramientas, leímos en el [sitio web de OmegaT](#) que los desarrolladores preferían que para localizarlo se usara el propio programa («Needless to say, we would prefer you to use OmegaT for the OmegaT localization work. You can however use other software if you wish»). Aunque no era una condición indispensable, nos pareció perfectamente lógico usar la propia herramienta. Además, cuanto más práctica adquiriéramos en el uso de esta aplicación, más fácil se nos haría localizarla y dar con traducciones más apropiadas o de mejor calidad.

Este factor condicionó que escogiéramos OmegaT también para localizar Ignuit: si íbamos a traducir dos programas, pensamos que sería mucho más sencillo y razonable emplear la misma herramienta en ambos casos.

Una cuestión importante a tener en cuenta era nuestra familiaridad con las herramientas disponibles. En el momento de empezar la asignatura TPD (Traducción de Productos Digitales), para la que se llevaba a cabo esta tarea, conocíamos varias herramientas, contando tanto aplicaciones de instalación local como recursos en la nube e incluyendo SGET, herramientas TAO, programas como editores de texto y repositorios en línea. Algunas de las aplicaciones las sabíamos usar; algunas simplemente sabíamos que existían; otras las descubrimos buscando en la red. Estábamos dispuestos a aprender el funcionamiento de una herramienta antes desconocida si veíamos que podía sernos de utilidad, siempre que el esfuerzo fuera proporcional.

La elección iba a estar en gran parte supeditada a los programas que fuéramos a localizar. Por ejemplo, si nos decidíamos por alguna de las opciones disponibles en Crowdin (como Wire, GDevelop o Code), una buena manera de llevar a cabo la localización habría sido usar directamente la misma plataforma.

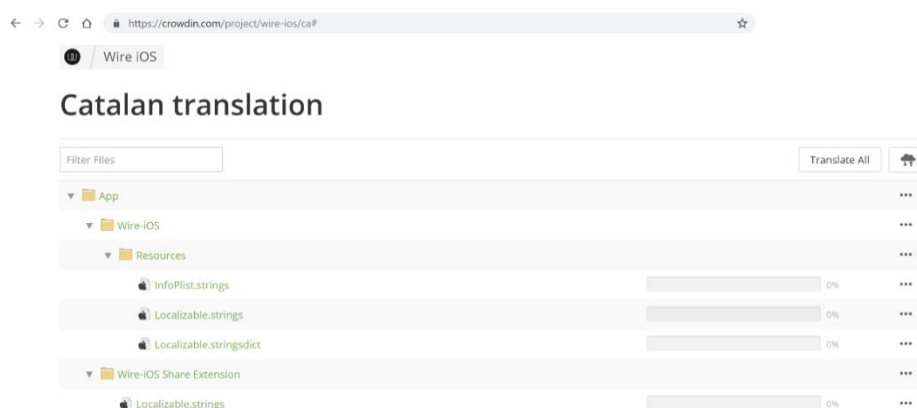


Ilustración 7. Interfaz de Crowdin. Carpetas de código de Wire iOS



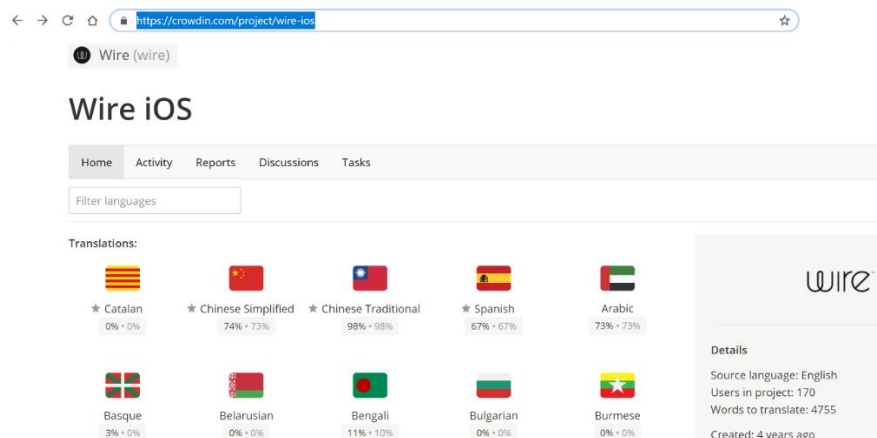


Ilustración 8. Interfaz de Crowdin. Algunas lenguas de localización disponibles

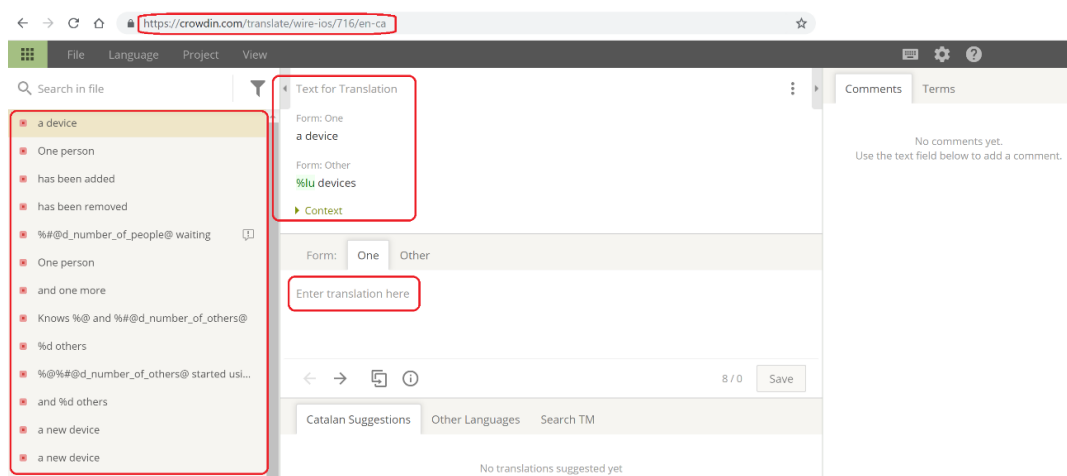


Ilustración 9. Interfaz de Crowdin. Cadenas de texto de Wire iOS

En las capturas de arriba vemos cómo habría sido la localización al catalán de la *app* Wire para iOS a través del editor en línea de Crowdin. En este caso, ni siquiera habría sido necesario descargar ningún archivo ni aplicar ningún tipo de filtro para reconocer las cadenas de los originales. En los proyectos de Crowdin, todo está bien preparado para que los traductores puedan simplemente abrir los archivos y traducir segmentos cómodamente.

Además de estos sitios web colaborativos en los que todo está convenientemente preparado —y que habrían tenido sentido sobre todo si decidiáramos traducir algo de lo que se proponía allí—, teníamos la opción de utilizar cualquier otra herramienta TAO o SGET. A parte de algunas aplicaciones que exponemos en detalle un poco más adelante, valoramos: Wordfast, Memsource, SmartCAT, Matecat, Pootle, Virtaal, Catalyst y Passolo.

Entre los criterios que influyeron en nuestra decisión está la especificidad de los programas a utilizar. Es decir, considerando que teníamos que localizar *software*, parecía natural hacerlo con un recurso diseñado para tal fin —como, por ejemplo, Catalyst o Passolo—. Aun así, hubo otros factores con más peso que nos hicieron decantar en contra:

- La falta de dominio: no conocíamos en profundidad el funcionamiento de estas herramientas.

- El hecho de que fueran *software* privativo para el que no teníamos licencia (una posible solución habría sido trabajar siempre desde la universidad, pero no lo vimos práctico en absoluto).

Aprovechando que hemos mencionado algunos de nuestros criterios, queremos expandir la lista nombrando aquí otros de los más importantes que intervinieron en la elección:

- Software* libre:

Este condicionante no solamente era un criterio propio, sino también un elemento indicado de manera explícita como positivo en las instrucciones de nuestro profesor.

-Herramientas que todos los miembros del grupo ya tuvieran instaladas o de fácil instalación (por razones pragmáticas obvias):

Desde un principio, teníamos claro que dedicaríamos horas a este trabajo tanto en la universidad como en casa. Así pues, debíamos buscar una metodología y unas herramientas que nos permitieran avanzar fácilmente como grupo y desde cualquier lugar.

-Opción práctica en general:

Lo más importante para nosotros era seleccionar los medios y estrategias que nos resultaran más sencillos a la par que productivos. Tanto en el método de trabajo como en las herramientas utilizadas, buscábamos fluidez y queríamos que nuestra inversión de tiempo y esfuerzo fuera proporcionada y razonable.

Este argumento está relacionado con lo mencionado arriba acerca de la falta de dominio; si bien estábamos más que dispuestos a dedicar tiempo a aprender, quisimos evitar que la herramienta de traducción escogida se acabara convirtiendo en un obstáculo.

No obstante, lo primordial era decidir qué íbamos a localizar. Puesto que finalmente nos decantamos por OmegaT e Ignuit, nuestras opciones se concretaron en unas pocas. OmegaT siempre fue vista como prioridad por lo expuesto al principio de este apartado, aunque antes quisimos poner algunas alternativas sobre la mesa.

Para poder escoger, lo primero que teníamos que hacer era observar el material con el que debíamos trabajar: básicamente, nos referimos a las extensiones y los formatos de los archivos traducibles. Evidentemente, el *software* de traducción que acabáramos usando tendríamos que soportar los tipos de archivos a traducir.

Hicimos una lista mental de todas las aplicaciones que conocíamos. Como hemos dicho antes, también buscamos en Internet nuevas opciones. Luego empezamos a investigar y a probar cada una de ellas para ver cuál podría ser la más apropiada para nuestro trabajo. Entre las operaciones que llevamos a cabo está la de crear un proyecto nuevo con nuestros archivos a traducir (*translatable*) en algunas herramientas para ver el comportamiento de cada una.

Nombramos algunas de las que probamos —además de OmegaT—: MemoQ, Trados, Déjà Vu, Lokalise, GTranslate, Poedit.

### 1.6.1. Otras herramientas TAO valoradas

Antes de posicionarnos por la herramienta TAO OmegaT, valoramos las siguientes herramientas:

- MemoQ

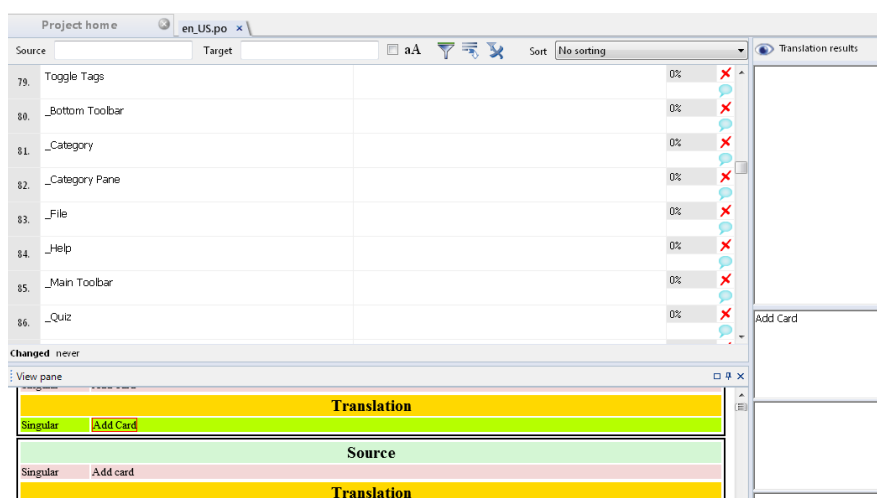


Ilustración 10. Interfaz de MemoQ. Archivo de interfaz de Ignuit

- Lokalise

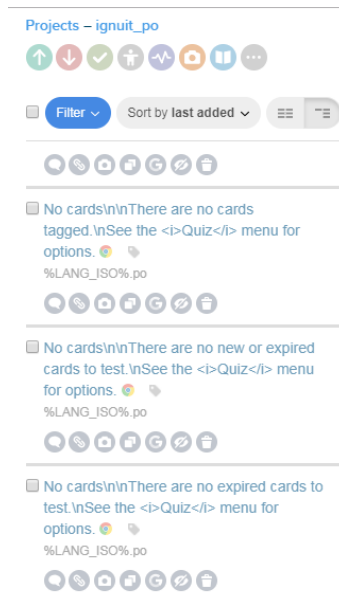


Ilustración 11. Interfaz de Lokalise. Archivo de interfaz de Ignuit

- Poedit

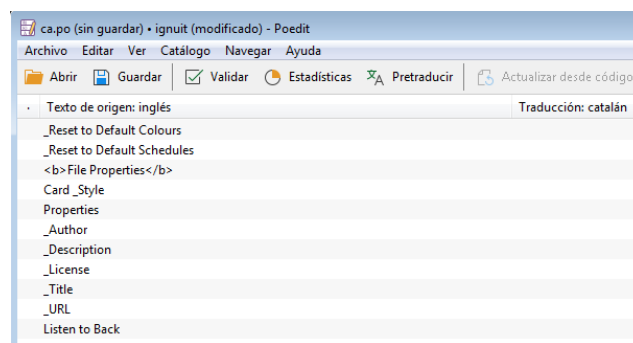


Ilustración 12. Interfaz de PoEdit. Archivo de interfaz de Ignuit

- GTranslate

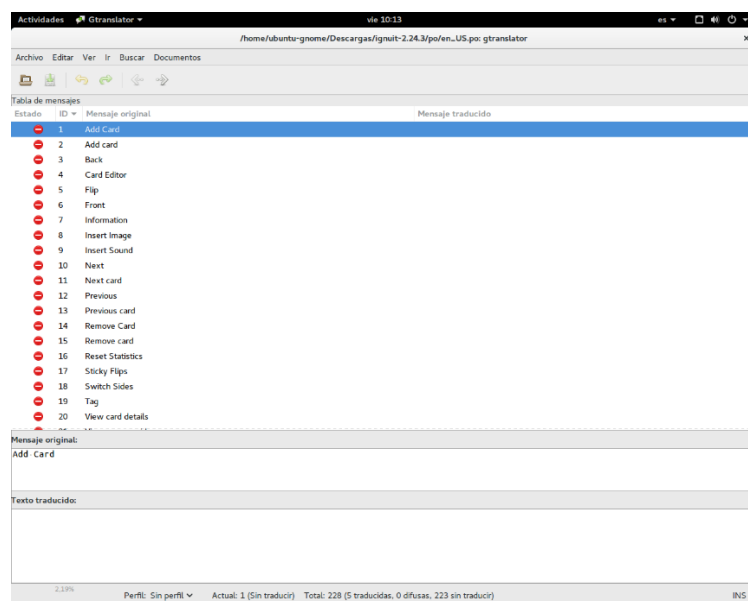


Ilustración 13. Interfaz de GTranslate. Archivo de interfaz de Ignuit

### 1.6.2. Herramientas complementarias

OmegaT fue el programa que usamos para lo que es estrictamente traducir y realizar el control de calidad. No obstante, como ya se sabe, los encargos de traducción conllevan muchas tareas más allá de la traducción y la revisión o el control de calidad. En la tabla que sigue, nombramos algunos otros recursos que utilizamos para llevar a cabo nuestro trabajo<sup>8</sup>:

Programa / recurso	Uso de la herramienta	Función
Telegram	Mensajería instantánea	Comunicación interna
Skype	Videollamadas	Comunicación interna
Yahoo Mail	Servidor de correo	Comunicación con desarrolladores y traductores Puente para la creación de cuentas de usuario en repositorios en línea
Google Drive	Almacenamiento de archivos en la nube	Gestión del proyecto
Hojas de cálculo de Google Drive	Hojas de cálculo en la nube	Preparación > gestión de terminología (glosario propio)
MemoQ	SGET	Análisis > recuento de palabras Análisis > análisis de formatos
MateCAT	SGET	Análisis > recuento de palabras
Memsource	SGET	Preparación > extracción de términos
Notepad++	Editor de textos	Análisis > análisis de formatos Postproducción > verificación técnica Preparación > lectura de archivos > glosarios
Microsoft Excel	Hojas de cálculo	Preparación > gestión de terminología Varias fases > lectura de archivos (ej.: glosarios, recuentos de palabras)
GitHub	Sistema de control de versiones en línea	Obtención de originales Gestión del proyecto Gestión de traducciones en equipo a través de un servidor Obtención del archivo TMX final
AntConc	Gestor de corpus	Preparación de materiales de referencia > extracción terminológica
Catalyst	SGET para localización	Postproducción > verificación técnica*
Vocab Grabber	Extractor de términos	Preparación de materiales de referencia > extracción terminológica**
Five Filters	Extractor de términos	Preparación de materiales de referencia > extracción terminológica**
Okapi Framework Olifant	Editor de memorias de traducción	Postproducción > extracción de segmentos de memoria de traducción

Tabla 2. Herramientas complementarias utilizadas

\*Probamos Catalyst para el testeo de las traducciones. No obstante, la falta de entrenamiento en el uso de este programa impidió que lográramos ver correctamente los archivos traducidos.

\*\*Las herramientas Vocab Grabber y File Filters no fueron del todo útiles en este proceso.

<sup>8</sup> No hacemos referencia aquí al proceso de traducción específicamente, sino al trabajo en su conjunto.

En un principio, quisimos gestionar el proyecto a través de la interfaz web de [Trello](#). Al poco tiempo vimos que con el resto de instrumentos de los que disponíamos, el uso de Trello pasaba a representar un trabajo de más innecesario.

## 2. Gestión del proyecto

### 2.1. Planteamiento del proyecto

Hemos mencionado antes que quisimos enfocar este trabajo como un encargo real, teniendo en cuenta unas fases concretas. Estas se detallan en *MTradumàtica i la formació de traductors en Traducció Automàtica Estadística* (Martín-Mor, A.; Piqué, R.: 2017) y en *La digitalización del proceso de traducción: tareas y herramientas* (Piqué, R.; Sánchez-Gijón, P.: 2006).

Nos situamos en la hipótesis de que éramos un grupo de traducción profesional que recibía dos encargos de localización por separado. Evidentemente, había diferencias entre un encargo y el otro, pero la intención era mantener una consistencia como grupo y una serie de criterios sólidos que se aplicaran a ambas tareas. Un ejemplo de ello es el seguimiento de las mismas guías de estilo o las mismas pautas de trabajo y funcionamiento de equipo tanto para OmegaT como para Ignuit.

Mediante estos encargos, fuimos partícipes de lo que son los proyectos colaborativos. En primer lugar, esta fue nuestra primera experiencia en el mundo del *software* libre de código abierto. Este proyecto nos ha permitido conocer mejor la ideología de la información libre. No solo esto, sino que ahora podemos decir que formamos parte de ello.

Nuestras aportaciones estuvieron sincronizadas a través de un servidor que nos permitía compartir contenidos y editar cadenas de texto cada uno por su cuenta pero sobre un mismo proyecto en OmegaT. Por otro lado, abordamos todo el trabajo, no solo la traducción, desde la mentalidad de la convergencia de ideas y la suma de potenciales; es decir, a pesar de tener unos roles asignados (los exponemos más adelante), cada integrante del grupo participó de todas las fases y en todos los aspectos relacionados de cada una, aportando todo lo posible y ayudando a los compañeros. Precisamente porque todos éramos *amateurs*, lo que planteamos fue un modelo de colaboración en el que llevamos a cabo tareas de investigación simultáneamente y todos hicimos un poco de todo, compartiendo y sumando ideas.

#### 2.1.1. Uso de la plataforma GitHub

Uno de los instrumentos que sin duda ha resultado clave en el proceso ha sido [GitHub](#). GitHub se utiliza principalmente para la creación de código fuente de programas de ordenador. Utiliza el sistema de control de versiones Git. Los sistemas de control de versiones (VCS, *version control systems*, en inglés) permiten mantener un historial de versiones de un archivo o conjunto de archivos, registrando todos los cambios y concediendo la posibilidad de recuperar dichos archivos en el estado de un punto concreto en el tiempo.

Los proyectos en GitHub pueden ser privados o públicos. Toda persona que tenga una cuenta de usuario puede o bien participar en proyectos creados por otros usuarios o bien crear de cero un nuevo repositorio. Los proyectos públicos están a la vista de todos, mientras que los privados solo son visibles por los usuarios aceptados por quien administre el repositorio. Es importante recordar que solo se puede subir a la plataforma contenido para el cual tengamos licencia. Es decir, incurriríamos en una infracción si publicáramos el código de un determinado programa sin tener los permisos correspondientes. También cabe decir que para crear un proyecto privado es necesario abonar una cuota.

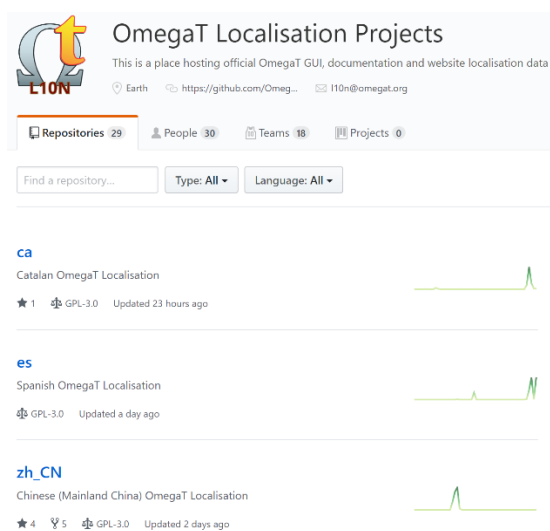


Ilustración 14. Proyectos de localización de OmegaT en GitHub

Ilustración 15. Creación de un repositorio en GitHub

La localización de OmegaT se administra a través de GitHub. Por tanto, no fuimos nosotros mismos quienes decidimos aprovechar este recurso, sino que la elección vino dada por las circunstancias. Desde el principio hasta el final, todas nuestras traducciones se añadieron al proyecto mediante la conexión de OmegaT con el repositorio virtual. Para Ignuit, empezamos traduciendo «en local», trabajando cada persona en una combinación lingüística distinta. No obstante, una serie de eventualidades nos acabaron llevando a crear nuestro propio repositorio en GitHub para Ignuit.

Los enlaces a los repositorios de OmegaT en GitHub para catalán y español son los siguientes:

<https://github.com/OmegaT-L10N/ca>

<https://github.com/OmegaT-L10N/es>

Veremos en el apartado [2.3.1](#) qué es exactamente lo que hicimos para sincronizar nuestras traducciones con la plataforma. También explicaremos en un apartado posterior nuestra experiencia en GitHub con Ignuit.

## 2.2. Roles y reparto del trabajo

Primero identificamos nuestras necesidades y establecimos en base a ello unas funciones para cada persona. Esto no impidió, tal como hemos dicho, que en la práctica los roles en cierta manera se fusionaran.

La asignación de tareas vino determinada por la experiencia previa y por preferencias personales.

Puesto que Mariona había trabajado antes en una agencia de traducción como gestora de proyectos, le atribuimos esta función. Por su parte, Moisés estaba interesado en la gestión terminológica, motivo por el cual él se encargó de extraer términos de los archivos traducibles de Ignuit y de sentar las bases de un glosario propio que entre todos editamos después.

La gestión del proyecto englobó responsabilidades como: detección de archivos traducibles, recuento de palabras, preparación de originales, identificación de tareas necesarias y establecimiento de fechas límite para completarlas, creación de carpetas de proyecto que se enviarían a los traductores (solo para Ignuit), recepción del material traducido y preparación de la entrega final.

Desde el comienzo, dejamos claro que todos queríamos traducir, fuera en mayor o menor medida, y que queríamos hacerlo con nuestras lenguas nativas como meta. En base a las orientaciones del profesor, marcamos las 2.000 palabras por persona como mínimo

aproximado. Para OmegaT, al no tener texto traducible al italiano, Eugenia y Tatiana se repartieron el español y consensuaron entre ellas en qué cadenas o archivos trabajaría cada una. Moisés y Mariona se repartieron el catalán para OmegaT y para Iguit, también estableciendo entre los dos dónde intervendría cada uno.

De nuevo por el hecho de tener más competencias y un especial interés, Moisés accedió a ser el encargado principal de las tareas de ingeniería, tales como el análisis de formatos, la verificación técnica o la instalación de Ubuntu e Iguit paralela a la traducción. Mariona colaboró como ingeniera en la detección de los archivos traducibles, la preparación de las carpetas de proyecto y la verificación técnica.

El proceso de revisión se compuso de: revisión individual, revisiones cruzadas y revisión conjunta (los pasos se detallan en el apartado [5.1](#)). En cuanto a las revisiones cruzadas, Eugenia se encargó de las traducciones de Tatiana; Moisés y Mariona revisaron sus traducciones mutuamente. La revisión del italiano solo pudo llevarse a cabo como parte de la revisión conjunta.

La comunicación con los desarrolladores —ya sea de OmegaT o de Iguit— fue una función correspondiente a Eugenia y Mariona.

La lista de tareas que ejecutó cada miembro se puede observar en el cuadro resumen ([1.1](#)) de la presente memoria. Cada uno de los pasos se explica como es debido en el apartado que le corresponde.

Para la comunicación interna, establecimos un código de nomenclaturas y colores para referirnos a cada integrante:

### 2.3. Flujo de trabajo

Presentamos aquí un esquema de la organización que siguió el grupo en el desarrollo del proyecto:

Fecha (2018)	Descripción de las fases principales
2 oct – 29 oct	Investigación previa. Establecimiento del marco teórico y las bases del trabajo. Formación del grupo y elección de los productos a traducir. Intento de contacto con algunos desarrolladores y grupos de traducción (ej.: Softcatalà).
29 oct – 6 nov	Análisis en profundidad de los originales (lingüístico y técnico). Detección de necesidades. Delimitación de tareas. Asignación de roles. Extracción de archivos traducibles (y recuento). Recuento de palabras. Recopilación y elaboración de materiales de referencia. Búsqueda de textos comparables (especialmente, para Iguit); observación de las traducciones de versiones anteriores de OmegaT. Intento de entrenamiento de un motor de TA. Asentamiento de las pautas formales y elección de guías de estilo. Descarga de diccionarios para la configuración del corrector de OmegaT. Consenso terminológico. Configuración de OmegaT según necesidades específicas del proyecto (para cada combinación lingüística). Creación de una cuenta de Yahoo Mail y de carpetas compartidas de Google Drive. Primer contacto con los desarrolladores y otros traductores.
7 nov – 26 nov	Creación de cuentas en GitHub.

	Adhesión al grupo de localizadores de OmegaT y a la lista de distribución de usuarios de OmegaT. Traducción.
27 nov – 4 dic	Revisión: individual, cruzada y conjunta. Ajuste de terminología.
4 dic – 11 dic	Comprobación técnica y testeo (y ajuste de la traducción revisada acorde con los resultados). Extracción de los materiales traducidos y las memorias de traducción resultantes en formato TMX. Redacción de la memoria escrita del trabajo.
Pendiente en el momento actual	Preparación de la presentación oral del trabajo. Alineación de las traducciones para lograr mayor coherencia. Mejora de las traducciones en general antes de su publicación. Aplicación de las traducciones hechas al software para su publicación y posterior aprovechamiento por parte de cualquier usuario.
Tareas transversales	Documentación. Comunicación con los desarrolladores.

Tabla 3. Flujo de trabajo

Algunas de las fases de la tabla de arriba se desarrollan en este mismo punto, mientras que otras se explican más minuciosamente en otros apartados.

Puesto que Tatiana y Eugenia no iban a trabajar en los mismos archivos de la traducción de OmegaT, las dos pudieron empezar su trabajo simultáneamente. Se dedicaron primero a esta herramienta, traduciendo ambas al español, y pasaron después a la traducción de Iguit —Tatiana en italiano y Eugenia en español—.

Moisés y Mariona sí iban a tener archivos en común y solo iban a traducir al catalán. Para evitar cualquier tipo de problema, creímos conveniente que uno de ellos empezara con Iguit y la otra persona con OmegaT.

Calculamos un número de segmentos aproximados y establecimos una fecha límite para la traducción, tras la cual se invirtieron los roles y se llevó a cabo la revisión cruzada.

### 2.3.1. Primeros pasos del proyecto OmegaT

Por el simple hecho de haber descargado y probado la versión 4.1.5 (2), nos habíamos dado cuenta de que la localización de la interfaz no estaba completa ni en español ni en catalán. Sin embargo, desconocíamos si en ese momento ya había alguien trabajando en ello. Además, según nos había hecho notar nuestro profesor, podía ser que los segmentos que quedaban por traducir no llegaran al mínimo de palabras que debíamos traducir para este proyecto. En cuanto al italiano, parecía que la interfaz estaba completamente traducida. En resumen, pues, teníamos que investigar en qué fase se encontraba la localización de OmegaT, no solo de la interfaz, sino también del resto de materiales —manuales de usuario, página web, etc.—.

La información sobre cómo localizar OmegaT la obtuvimos, principalmente, de la web [How To – Translating OmegaT into your language](#). Siguiendo las instrucciones de la mencionada web, nos unimos a las listas de distribución de localizadores de OmegaT ([OmegaT /10n](#)) y de usuarios de OmegaT ([OmT](#)). Asimismo, contactamos con el desarrollador, Didier Briel, y con el actual gestor de localizaciones, Kos Ivantsov, quienes nos explicaron los pasos que debíamos seguir a continuación —esto es, proporcionarles nuestros nombres de usuario de GitHub para que nos incluyeran como colaboradores del repositorio—.

Nuestro compañero de clase, Marc Riera, tuvo un papel fundamental en el desarrollo del proyecto, especialmente en esta primera fase. Marc había traducido ya, de manera voluntaria, muchas cadenas de texto de OmegaT, con lo cual tenía más por la mano el



funcionamiento de la herramienta y de GitHub. Podemos decir que él fue nuestro guía en cuanto al uso de OmegaT y la localización del programa. Marc nos indicó, por otra parte, que el elemento principal de la localización de OmegaT era el archivo «Bundle.propiedades», correspondiente a las cadenas de la interfaz, o *GUI*, por sus siglas en inglés.

En un primer momento, también establecimos contacto con uno de los traductores de una versión anterior de OmegaT. Respondió a nuestro mensaje y aceptó ofrecer su ayuda, si bien no hizo falta recurrir a él posteriormente.

Una vez aceptados como colaboradores en GitHub, solo hizo falta abrir OmegaT y usar la función «Descargar proyecto en equipo», tal como se muestra en la captura (en inglés):

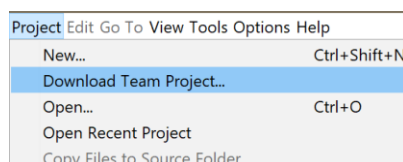


Ilustración 16. Interfaz de OmegaT. Descarga de proyecto en equipo

A continuación, solo hace falta introducir el enlace al repositorio en línea y seleccionar una carpeta local en la que se guardarán todos los archivos que componen el proyecto. En este caso, debemos crear una carpeta nueva.

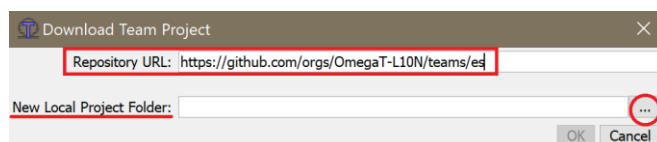


Ilustración 17. Conexión de OmegaT con un repositorio en línea

La primera vez que se realiza este proceso, el programa tarda cierto tiempo en descargar todo el contenido. Finalmente, se abre una ventana que muestra los archivos traducibles, mientras que en la ventana principal de OmegaT se abre el archivo principal: «Bundle.properties», listo para su traducción.

Después de haberlo descargado una primera vez, siempre que queramos volver a abrir ese mismo proyecto, podemos llegar a él con la opción «Abrir proyecto reciente» («Open Recent Project» en la captura):

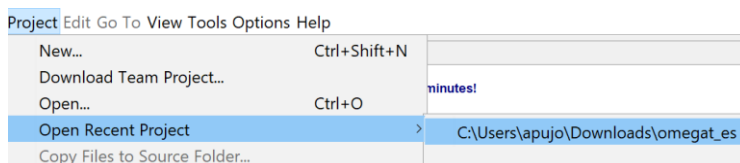


Ilustración 18. Interfaz de OmegaT. Abrir proyecto reciente

Hace falta estar conectado a Internet para que las modificaciones que realicemos se sincronicen con el repositorio. Si no es el caso, se pueden guardar en la carpeta local y sincronizarse en otro momento. Aun así, en ocasiones, teniendo acceso a Internet, aparece el mensaje «No hay conexión disponible. El proyecto está en modo sin conexión». Podemos solucionar el problema actualizando el proyecto (F5) hasta que aparezca el mensaje «Conexión disponible. El proyecto está en modo de proyecto de trabajo en grupo».

Gracias a este sencillo método, todos pudimos acceder al control de versiones de la localización de OmegaT al catalán y al español y editar cadenas de texto.

Cuando se trabaja con OmegaT, cada dispositivo tiene asignado un nombre de usuario o ID. Para participar de las traducciones en equipo a través de GitHub, es aconsejable que si una misma persona trabaja desde ordenadores diferentes, edite la ID del equipo que esté usando para que sea siempre la misma, a fin de identificar con facilidad el autor de cada

modificación. Esta opción se encuentra bajo «Opciones > Preferencias > Trabajo en grupo» (en la captura, «Team»).

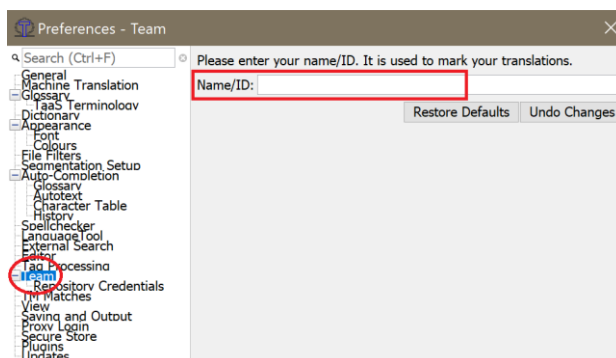


Ilustración 19. Interfaz de OmegaT. Ventana de preferencias. ID del traductor

Un gran problema al que nos enfrentamos y que queremos comentar es que en varias ocasiones al abrir el proyecto en equipo la memoria de traducción («project\_save.tmx») aparecía vacía —todas las traducciones, incluso las que ya existían antes de nuestra participación, habían desaparecido—. Desconocemos si el fallo es generado por GitHub o por OmegaT. Intercambiando correos con Kos Ivantsov, descubrimos que esta incidencia no solo nos había ocurrido a nosotros. Para recuperar el proyecto, fuimos al historial de versiones en GitHub (apartado «Commits») y buscamos el «project\_save.tmx» en el estado que al que queríamos restaurar el proyecto. Descargamos esa memoria TMX y la introdujimos en la carpeta «tm\auto». Al actualizar el proyecto en OmegaT (F5), las traducciones se habían autopropagado desde la versión restaurada.

### 2.3.2. Dos fases diferentes para el proyecto Ignuit

Inicialmente, empezamos a traducir desde una carpeta local. Para el español y el italiano, se encargaba una sola persona de traducir los archivos; para el catalán, teníamos a dos traductores, pero uno de ellos empezaría con OmegaT. Moisés fue quien empezó por Ignuit, mientras que Mariona empezó por OmegaT. La intención era que cuando cada uno acabara su trabajo, enviaría la carpeta del proyecto comprimida a quien correspondiera, fuera para su revisión o, en el caso del catalán, para continuar la traducción.

El problema llegó al presentarse un contratiempo totalmente inesperado y ajeno a nuestro control: uno de nuestros ordenadores personales dejó de funcionar. Por fortuna, la última copia de seguridad era de 4 días antes y estaba colgada en la nube, por lo que la cantidad de material que se perdió fue nimia. Sin embargo, esto nos hizo replantear la situación, hasta que dimos con la idea de crear nuestro propio repositorio en GitHub para el proyecto Ignuit. Hacerlo así nos facilitaría, por un lado, el acceder al contenido de cualquiera de las tres combinaciones lingüísticas, independientemente del ordenador con el que trabajáramos. Esto nos daba acceso al «project\_save.tmx» actualizado de todas las lenguas para consultar las traducciones hechas por los compañeros hasta entonces. En segundo lugar, nos evitaba el paso de comprimir y enviar las carpetas del proyecto cada vez que quisiéramos compartir nuestros avances. Por otra parte, con el historial del control de versiones teníamos una visión de todos los cambios efectuados sobre el proyecto.

Esta decisión presentaba un inconveniente: para hacer público el proyecto en GitHub necesitábamos autorización de Tim Musson y debíamos tener claro que cualquier individuo podía acceder a esa información. Si, de lo contrario, creábamos un repositorio privado, teníamos que pagar el importe requerido. La primera opción habría sido viable, ya que Tim Musson está conforme en la publicación del repositorio. A pesar de ello, acabamos optando por hacerlo privado, al menos por el momento, para tener un control sobre los usuarios que podían editar las cadenas de texto —es decir, nosotros cuatro—.

La cuota de creación de repositorios privados no es elevada y se paga una vez al mes. Con este pago, un mismo usuario puede crear cuantos repositorios quiera. En el momento de realizar esta acción, faltaba menos de un mes para la entrega del presente trabajo, con lo cual solo hacía falta abonar el importe una sola vez y, entregado el trabajo, podíamos cancelar la suscripción.

Primero creamos tres repositorios, uno para cada lengua meta. Algo más tarde —por motivos que se explican en el apartado [3.3.1.2](#), relacionados con la autopropagación de nuestras traducciones—, añadimos un cuarto para el archivo «Ignuit.pot» (EN>ES).

Existe la opción de permitir el acceso a otros usuarios en un futuro próximo. De hecho, Tim Musson querría que su programa se adaptara a nuevos *locales*. El uso de un sistema de control de versiones sería una buena opción para gestionar la localización colaborativa de este *software* a todas las lenguas que se desee.

### 3. Fase de preparación

#### 3.1. Introducción al análisis del TO

Creemos que para realizar un buen análisis de traducción, partiendo de que en este caso nos enfrentamos a la especialidad de localización, es necesario tener en cuenta dos aspectos: el análisis del programa y el análisis lingüístico. Estos dos, complementados, vienen a ser el análisis del TO (texto original).

En lo que se refiere al programa, analizamos su funcionalidad —para qué se desarrolló— y su estructura —qué elementos lo constituyen: ventanas, paneles, botones...—. En relación con estos elementos, intervienen tanto el código como los formatos utilizados en el diseño del programa.

El aspecto lingüístico proporciona significado a la funcionalidad y la estructura del programa.

La localización de programas informáticos tiene una serie de especificidades que hacen que la tarea del traductor (localizador) sea muy distinta a la de otras especialidades. Para empezar, el TO posee unas características claramente diferenciadas de un TO de cualquier otro campo. En segundo lugar, se trabaja con una enorme variedad de formatos de archivo. Entre otros factores, también es esencial tener en cuenta la herramienta TAO que se utiliza, sobre todo en lo que se refiere a las reglas de segmentación.

La [internacionalización](#)<sup>9</sup> supone un paso previo muy determinante.

En localización, es importante tener en cuenta varios factores: el propiamente lingüístico —común a todas las especialidades de traducción—, el visual —elemento esencial—, el cultural —adaptación a un *locale*, con todas las implicaciones que ello supone— y el técnico —una faceta vinculada sobre todo a la ingeniería de la localización—.

Podemos decir que las dos aplicaciones que hemos localizado estaban convenientemente adaptadas para su localización, es decir, internacionalizadas. Sobra decir que esto ha facilitado enormemente nuestra intervención.

Desde una perspectiva general, ambos programas presentan funcionalidades diferentes: el primero es una herramienta TAO, mientras que el segundo es una aplicación de asistencia a la memorización mnemotécnica. El primero consta de varias herramientas que el traductor/localizador empleará para desarrollar su labor. Por otra parte, el segundo cuenta con herramientas que puede utilizar cualquier usuario —con mínimos conocimientos sobre Linux—.

---

<sup>9</sup> GALA. Globalization & Localization Association (2018). *What is internationalization?* [en línea]: «*Internationalization is a design process that ensures a product (usually a software application) can be adapted to various languages and regions without requiring engineering changes to the source code*».

### 3.2. Análisis lingüístico de los originales

En lo que se refiere a la lingüística, una dificultad que en general afecta a cualquier localizador de *software* es la comprensión del original. Nos enfrentamos, normalmente, a frases cortas e incluso palabras sueltas cargadas de significado y con muchos términos técnicos.

La dificultad en el caso de OmegaT era relativa, puesto que las convenciones son más rígidas: contábamos ya con una memoria de traducción y con el estilo aplicado en traducciones anteriores. Fue de gran ayuda, en este sentido, que la interfaz de usuario del programa estuviera ya traducida al italiano y prácticamente completa al 100% en español. Las memorias de traducción (MT) de versiones anteriores del *software* se encontraban por defecto en la subcarpeta «tm\auto» del proyecto.

En cuanto a Ignuit, disponíamos de la traducción de la interfaz al alemán y al ruso, pero no todos teníamos conocimiento de estas lenguas y, por otra parte, era nuestro primer contacto con la aplicación, con lo que no entendíamos algunas de las funciones o algunos de los botones, menús, etc. Aun así, cabe decir que no es una interfaz compleja; se ha diseñado para que cualquier usuario pueda sacar provecho del programa.

Una ventaja que tuvimos fue la alta «objetividad» de los textos con los que trabajábamos, en contra de lo que podría haber ocurrido al elegir otro tipo de producto, como una página web o un videojuego. Es decir, no tuvimos que lidiar con problemas que se suelen presentar en otros tipos de texto, entre las que podríamos nombrar, por ejemplo, dificultades de carácter creativo, expresiones polisémicas, nombres de personajes, etc.

Respecto a la terminología, profundizaremos más en la extracción de términos en el apartado «Preparación de materiales de referencia» (3.4).

En nuestro primer análisis, durante las fases de elección de *software* a traducir y preparación de originales, detectamos ya algunas dificultades concretas. No obstante, fue en las fases de traducción y revisión cuando extrajimos la mayor parte de ellas. Las exponemos en los apartados correspondientes a la traducción y la revisión, ofreciendo aquí solo algunos ejemplos de las que detectamos a priori:

Característica	OmegaT	Ignuit
Formas de tratamiento	«You» formal en la interfaz. «You» informal en los manuales de ayuda.	Informal, tanto en la interfaz como en la documentación.
Estilo general de las oraciones/frases	Frases simples, generalmente cortas (sobre todo en la interfaz).	Frases simples, generalmente cortas (sobre todo en la interfaz).
Terminología específica o comprensión del original	Algunos ejemplos: « <i>pinpoint alignment</i> », «RTL» y «LTR».	Algunos ejemplos: « <i>sticky flips</i> », « <i>drill mode</i> », « <i>gtk</i> ».
Falta de contexto*	Ej.: « <i>Select All</i> »: «Seleccionar todo/a/os/as».	Ej. 1: « <i>Tag</i> »: ¿verbo o sustantivo? Ej. 2: « <i>Back</i> »: ¿Se refiere a « <i>Back of the card</i> » o a « <i>Go back</i> »?
Elección de términos traducidos	Debíamos observar versiones anteriores del programa o traducciones a otras lenguas para obtener	Debíamos consensuar entre todos las traducciones. Ej. 1: « <i>flaschard</i> »: tarjeta/ficha;

	la respuesta.	tarjeta/fitxa; scheda/flashcard.  Ej. 2: «Add card»: añadir/agregar/crear/[sin verbo] nueva tarjeta (y sus equivalentes en CA e IT).  Ej. 3: «front» y «back»: ¿Traducción más apropiada?
Presencia de <a href="#">hotkeys</a> <sup>10</sup>	Se marcan con «&».	Se marcan con «_».
Segmentos con estructuras paralelas; frases repetitivas	«An error occurred when loading the source files»; «An error occurred when saving the TMX».	«A value of 0 disables automatic playing»; «A value of 1 plays fronts automatically»; «A value of 2 plays backs automatically».
Referentes claros; bajo nivel de sinonimia	Ej.: «source file» y «target file», «segment».	Ej.: «card», «category», «box», «quiz».
Identificación de etiquetas	Es muy importante detectar determinadas etiquetas que OmegaT no filtra de manera automática**.  Ej.: <code>&lt;/b&gt; + texto traducible + &lt;/b&gt;</code>	

Tabla 4. Dificultades detectadas en el análisis del TO

\*Para algunas cadenas de texto (*strings*), identificamos el lugar en el que aparecían en el TO, pero no siempre fue así.

\*\*A través de «Options» > Preferences > Tag processing > Regular expression for custom tags» se pueden ajustar los parámetros relativos a las etiquetas para «bloquearlas» y así evitar que se desconfiguren al traducir.

Algo muy importante que debemos destacar es que las soluciones adoptadas no necesariamente tenían que ser las mismas en todas las lenguas meta. Aun así, siempre fue de vital importancia para todos mantener la consistencia como grupo y no usar fórmulas o términos muy dispares entre ellos. Debíamos ser coherentes y mantener una cierta homogeneidad como equipo, cosa que era buena no solo para nosotros, sino para el resultado final del trabajo.

### 3.3. Análisis técnico de los originales

El análisis técnico comprende la identificación de los archivos traducibles, el recuento de palabras o segmentos a traducir, nuestra familiaridad con los materiales traducibles y los tipos de archivo soportados por la herramienta TAO.

Una de las dificultades que tuvimos en las primeras fases de este trabajo fue detectar los archivos traducibles de los programas escogidos —en particular, los de Ignuit, porque partíamos de una carpeta de descarga que contenía mucho más que materiales a traducir, y algunos de los ficheros no sabíamos muy bien lo que eran—. Gran parte del proceso consistió en investigar y documentarnos para adquirir más conocimientos sobre formatos, extensiones de archivos, programación, sistemas operativos o informática en general.

<sup>10</sup> Caracteres que activan un determinado comando o botón, de manera similar a como actúan los atajos de teclado.

### 3.3.1. Identificación de los archivos traducibles

#### 3.3.1.1. OmegaT

En el caso de OmegaT, el procedimiento fue relativamente sencillo, puesto que desde un primer momento recibimos indicaciones por parte de los responsables de la localización del programa. Como línea general, seguimos las instrucciones de la web [How To – Translating OmegaT into your language](#), a pesar de que pudimos omitir algunos pasos, puesto que ahora la localización de OmegaT se gestiona a través de repositorios en línea y hay algunos procesos para los que no seguimos exactamente las pautas de dicha web.

Tal como vemos en esta página, son varios los materiales que podrían necesitar traducción a una determinada lengua. Veamos las siguientes capturas:

#### The translation work

At the time of writing, the material to be translated comprises:

- \* The user interface (menus etc.)
- \* The user manual (very comprehensive)
- \* The Instant Start guide (a short introduction to using OmegaT)
- \* The readme file (a document that contains general information about OmegaT)
- \* The Beginner's Tutorial
- \* The OmegaT web pages (probably where you are reading this)

Translators are free to tackle as much or as little of the work as they wish. In particular, the main user manual is likely to take around two weeks to translate, and several translators may wish to share the work.

Ilustración 20. Sitio web de OmegaT. Archivos traducibles de OmegaT

4. Download the files for translation.

For the **OmegaT program and documentation**, these are either the **"Full" package** or the **"Minimal" package**.

The full package contains the user manual, the instant start manual, the user interface resource bundle and the readme file. The minimal package contains the same files as the full package, except for the main user manual. You do not have to translate a complete package; you can, for instance, translate just the user interface (Bundle.properties), or just the instant start tutorial.

If you wish to translate the **OmegaT web pages**, download the **website localization package**. (Note: these are the English files. If you prefer to translate from a different language into which OmegaT has already been localized, please contact the **OmegaT webmaster**.)

Localizations of the OmegaT user interface and documentation are generally updated in step with new releases of OmegaT.

The OmegaT website includes files in HTML and, as of 2011, PHP format. To translate the latter in OmegaT, you must register the .php extension with the HTML and XHTML filter as follows:

Options > File Filters > Select "HTML and XHTML files" > Edit > Add > Then enter "\*.php" in the "Source Filename Pattern" column.

Finally, there is the **Beginner's Tutorial**, available [here](#).

Ilustración 21. Sitio web de OmegaT. Instrucciones para localizadores

Como vemos, toda persona que quiera aportar sus traducciones puede asumir el volumen de trabajo que desee.

El paso 4 de la imagen anterior no fue necesario, ya que al abrir el proyecto en grupo disponible en GitHub, todos los *translatables* se incorporan a nuestra subcarpeta local «source». En un primer estadio, sí que descargamos el contenido de lo que se denomina «*full package*», y nos sentimos algo perdidos porque no supimos identificar qué era lo que nosotros, como traductores, debíamos tocar. Al recibir noticias poco después con toda la información relativa a GitHub, nuestras dudas quedaron resueltas.

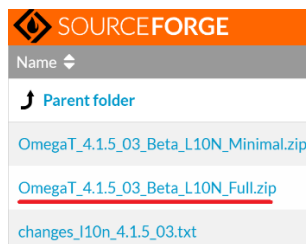


Ilustración 22. «Full package» en Source Forge de la localización de OmegaT

Una vez abrimos en OmegaT el proyecto en equipo sincronizado con GitHub, aparece automáticamente la ventana «Project Files» en la que se muestran todos los archivos sujetos a traducción. Eran 73 para la versión 4.1.5 (2), y podemos ver la mayoría de ellos en la captura que sigue:

Filename	Filter	Encoding	Segments	Unique segments
bundle.properties	Java(TM) Resource Bundles	US-ASCII	1,899	1,722
doc_src/AboutOmegaT.xml	DocBook	UTF-8	56	56
doc_src/App_Aknowledgements.xml	DocBook	UTF-8	15	15
doc_src/App_Keyboard.xml	DocBook	UTF-8	60	60
doc_src/App_Languages.xml	DocBook	UTF-8	562	549
doc_src/App_LanguageTool_plugin.xml	DocBook	UTF-8	20	20
doc_src/App_LegalNotices.xml	DocBook	UTF-8	22	18
doc_src/App_Scripting_plugin.xml	DocBook	UTF-8	40	34
doc_src/App_ShortCutCustomization.xml	DocBook	UTF-8	335	291
doc_src/App_TeamProjects.xml	DocBook	UTF-8	102	101
doc_src/App_Tokenizer_plugin.xml	DocBook	UTF-8	23	20
doc_src/App_Website.xml	DocBook	UTF-8	44	44
doc_src/Dictionary.xml	DocBook	UTF-8	32	31
doc_src/EditingBehavior.xml	DocBook	UTF-8	68	68
doc_src/FileFilters.xml	DocBook	UTF-8	159	146
doc_src/FilesAndFolders.xml	DocBook	UTF-8	156	150
doc_src/FilesToTranslate.xml	DocBook	UTF-8	146	145
doc_src/FormattedText.xml	DocBook	UTF-8	118	114
doc_src/Glossaries.xml	DocBook	UTF-8	88	87
doc_src/InstallingAndRunning.xml	DocBook	UTF-8	353	322
doc_src/InstantStartGuide.xml	DocBook	UTF-8	47	47
doc_src/Issues.txt	Text	windows-1252	291	289
doc_src/MachineTranslation.xml	DocBook	UTF-8	104	94
doc_src/Menu.xml	DocBook	UTF-8	463	333
doc_src/Miscellaneous.xml	DocBook	UTF-8	98	85
doc_src/OmegaUsersManual_xinclude_full.xml	DocBook	UTF-8	6	5
doc_src/PlainText.xml	DocBook	UTF-8	33	31
doc_src/ProjectProperties.xml	DocBook	UTF-8	63	55
doc_src/Regexp.xml	DocBook	UTF-8	167	145
doc_src/SearchAndReplace.xml	DocBook	UTF-8	16	15
doc_src/Searches.xml	DocBook	UTF-8	83	81
doc_src/Segmentation.xml	DocBook	UTF-8	86	80
doc_src/Spellchecker.xml	DocBook	UTF-8	43	42
doc_src/Tab5.xml	DocBook	UTF-8	18	16
doc_src/TranslationMemories.xml	DocBook	UTF-8	207	201
doc_src/UserInterface.xml	DocBook	UTF-8	329	304
readme_ca.txt	Text	UTF-8	91	81
release/win32-specific/CustomMessages.ini	Key=Value Text	windows-1252	5	5
scripts/check_rules.properties	Java(TM) Resource Bundles	US-ASCII	43	40
scripts/nbsp.properties	Java(TM) Resource Bundles	US-ASCII	2	2
scripts/replace_strip_tags.properties	Java(TM) Resource Bundles	US-ASCII	4	3
scripts/search_replace.properties	Java(TM) Resource Bundles	US-ASCII	6	3
scripts/spellcheck.properties	Java(TM) Resource Bundles	US-ASCII	26	14
scripts/strip_bid_marks.properties	Java(TM) Resource Bundles	US-ASCII	4	3
scripts/trim_cleanup_selected.properties	Java(TM) Resource Bundles	US-ASCII	11	11
scripts/tagwipe.properties	Java(TM) Resource Bundles	US-ASCII	34	33
website/404.html	HTML and XHTML	windows-1252	3	3
website/about.html	HTML and XHTML	windows-1252	25	25
website/contact.html	HTML and XHTML	windows-1252	19	19
website/documentation.html	HTML and XHTML	UTF-8	99	90
website/download.html	HTML and XHTML	windows-1252	137	118
website/howto/compatibility.html	HTML and XHTML	UTF-8	140	134
website/howto/compatibility_deja_vu.html	HTML and XHTML	UTF-8	27	21
website/howto/compiling_from_source.html	HTML and XHTML	windows-1252	55	52
<b>Total number of segments</b>				<b>8,402</b>
<b>Number of unique segments</b>				<b>7,731</b>
<b>Translated unique segments</b>				<b>1,913</b>

Ilustración 23. Interfaz de OmegaT. Ventana de archivos traducibles

Como podemos observar, son muchos los archivos traducibles. Son exactamente los mismos para el español que para el catalán, con la diferencia de que en español hay un gran número de segmentos ya traducidos. Nuestro objetivo primordial era completar el «Bundle» en catalán, porque esta es la prioridad de los desarrolladores, según nos comunicó Marc Riera. A partir de allí, intentaríamos contribuir todo lo posible en el resto de archivos. Para escoger qué otros materiales traduciríamos, echamos un vistazo a su contenido y nos decantamos por aquellos que parecían contener las cadenas que el usuario vería primero o a las que tendría que recurrir más a menudo.

En la columna «filename» figuran los nombres de las carpetas en las que se encuentra cada archivo. Todo lo que aparece inmediatamente antes de la extensión de los archivos es propiamente el nombre de cada uno de ellos. Por ejemplo, «doc\_src.AboutOmegaT.xml» es el archivo «AboutOmegaT», situado dentro de la carpeta «doc\_src» y con extensión «.xml».

El archivo de nombre «Bundle.properties», que, como hemos dicho, corresponde a la interfaz de usuario, aparece por defecto en primer lugar. En adelante, nos referimos a este como el «Bundle».

Dentro de la carpeta «source» que se crea automáticamente en nuestro sistema, vemos precisamente esta estructura. Es similar a la del «full package» que habíamos descargado antes, salvo algunas diferencias. Las más destacables es que en el «full package»



algunos elementos están renombrados como «.ignoreme» y que para GitHub se crea una carpeta llamada «.repositories». Los archivos «README.md» y «LICENSE» son importantes cuando creamos repositorios en GitHub.



Ilustración 24. Carpetas de archivos de OmegaT

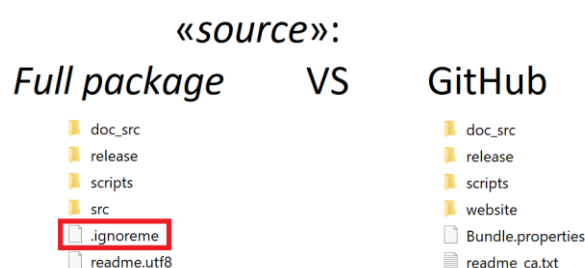


Ilustración 25. Archivos de la subcarpeta «source»

Los archivos traducibles de OmegaT pueden tener varios tipos de extensiones: «.properties» para la interfaz de usuario, «.xml» para los archivos del manual de ayuda, «.html» para la página web, y otras extensiones como «.ini» para la ventana de configuración de lengua que se ejecuta al instalar OmegaT o «.txt» para archivos como «Readme».

La extensión «.properties» se usa habitualmente en Java para contener cadenas de texto de *software* internacionalizado.

```

Bundle_ca.properties
79 # 0: Application name ("OmegaT")
80 # 1: Version ("3.5.1")
81 # 2: Update ("1")
82 app-version-template-pretty-update={0}
83
84 # 0: Application name + version
85 COMMAND_LINE_HELP=Usage: java -jar Omeg.
86
87 TEAM_TOOL_HELP=Usage: java -jar OmegaT.
88
89 # 0: Source language ("en", etc.)
90 # 1: Target language ("es", etc.)
91 TEAM_TOOL_INIT_COMPLETE=S'ha iniciat el
92
93 # buttons
94 BUTTON_OK=&d'acord
95
96 BUTTON_CANCEL=&Cancel\u00b71a
97
98 BUTTON_CLOSE=&Tanca
99
100 BUTTON_FILTER=&Filtra
101
102 BUTTON_REMOVEFILTER=&Elimina el filtre
103
104 BUTTON_REPLACE=&Substitueix
105 BUTTON_REPLACE_ALL=&Substitueix-ho tot
106
107 BUTTON_HOME=&Continu&ut
108
109 BUTTON_BACK=&Enrere

```

Ilustración 26. Estructura de un archivo de OmegaT con extensión «.properties»

No encontramos ningún archivo con extensión «.php», aunque la web [How To – Translating OmegaT into your language](#) sí que se menciona este tipo de extensión. Pensamos que quizá esto se aplica a versiones anteriores de OmegaT.



Evidentemente, todos estos archivos ya están preparados para ser traducidos con OmegaT. No nos hizo falta añadir ningún filtro especial ni realizar ninguna conversión de formato. La única dificultad que tuvimos es la de interpretar determinadas partes del «Bundle», como, por ejemplo:

```

Editor - Bundle.properties
-h, --help
    Show usage information.

--remote-project
    Download an omegat.project file from the URL given as <project
    path> and load the project.

--mode=[console-translate|console-pseudotranslatetmx|console-align]
    Specify a mode other than default (GUI).

See below for details.

--config-file=<path>
    A Java .properties file from which to load default settings.

--resource-bundle=<path>
    A Java .properties file to use for interface text
--config-dir=<path>
    The directory to read/write OmegaT configuration files from/to.

--disable-project-locking
    Do not attempt to lock the project's omegat.project file.

--disable-location-save
    Do not remember the last directory opened in the file picker.

--no-team
    Disable team project functionality.

```

Ilustración 27. Dificultad de identificación de texto traducible en «Bundle.properties»

En este caso concreto, usamos como referencia el «Bundle» ya traducido al español. Los archivos que finalmente modificamos fueron:

ES		CA
Tatiana	Eugenia	Traducción: Mariona; revisión: Moisés*
Resources.html	UserInterface.xml	Bundle.properties
Spelling.html		AboutOmegaT.xml
Glossaries.xml		InstantStartGuide.xml

Tabla 5. Archivos traducidos de OmegaT

\*La falta de tiempo impidió que Moisés tradujera cadenas de texto. Aun así, juntando las traducciones que hicimos al catalán entre OmegaT e Ignuit, superamos las 2.000 palabras por persona que nos habíamos marcado como objetivo.

### 3.3.1.2. Ignuit

El caso de Ignuit fue un poco más complejo. La carpeta comprimida que se descarga para instalar el programa contiene también todos los archivos traducibles —las cadenas de texto de la interfaz de usuario, archivos de ayuda como «README» y otros archivos en los que se explica cómo compilar, instalar o usar Ignuit—. Esta es la estructura de carpetas y archivos que compone todo el material necesario para ejecutar el programa.

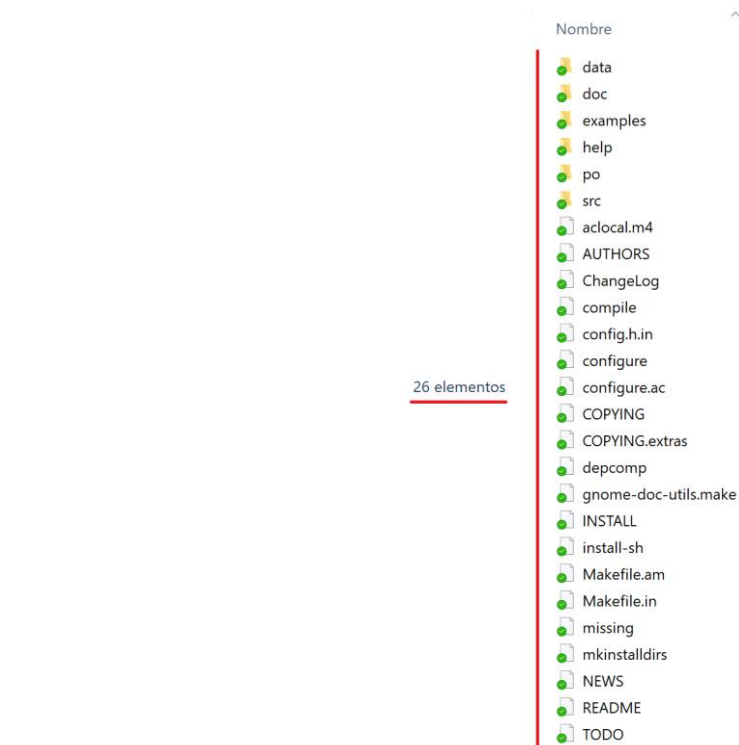


Tabla 6. Elementos de la carpeta de descarga de Ignuit

Nuestra pregunta, en un principio, fue: ¿Cuáles, de todos los archivos que encontramos en estas carpetas, son realmente los traducibles?

El esqueleto del directorio, junto con los nombres de los archivos, nos daban algo de información sobre cada uno de ellos. Podíamos suponer, por ejemplo, que en «INSTALL» se explicaba el proceso de instalación del programa, que la carpeta «data» contenía materiales necesarios para su ejecución, o que dentro de la carpeta «doc» había materiales de documentación sobre Ignuit.

En primer lugar, probamos a introducir la carpeta de descarga entera en la carpeta «source» de un proyecto nuevo en OmegaT. Este fue el resultado:

Project Files (11)

Filename	Filter	Encoding	Segments	Unique segments
ignuit-2.24.3/po/en_US.po	PO	UTF-8	243	237
ignuit-2.24.3/doc/texinfo.tex	LaTeX	windows-1252	1,417	1,083
ignuit-2.24.3/help/C/ignuit.xml	DocBook		395	363
ignuit-2.24.3/data/icons/scalable/actions/card-flip.svg	SVG Image and Adobe Illustrator Export	UTF-8	1	1
ignuit-2.24.3/data/icons/scalable/actions/card-known.svg	SVG Image and Adobe Illustrator Export	UTF-8	1	1
ignuit-2.24.3/data/icons/scalable/actions/card-unknown.svg	SVG Image and Adobe Illustrator Export	UTF-8	1	1
ignuit-2.24.3/data/icons/scalable/actions/quiz-begin.svg	SVG Image and Adobe Illustrator Export	UTF-8	1	1
ignuit-2.24.3/data/icons/scalable/apps/ignuit.svg	SVG Image and Adobe Illustrator Export	UTF-8	2	2
ignuit-2.24.3/examples/csv_import_test.csv	Magento CE Locale CSV	UTF-8	25	25
ignuit-2.24.3/po/ru.po	PO	UTF-8	270	35
ignuit-2.24.3/po/de.po	PO	UTF-8	241	0

Ilustración 28. Interfaz de OmegaT. Archivos reconocidos como traducibles

Nos pareció intuitivamente que debía de haber algo más.

Nuestra siguiente tentativa consistió, simple y llanamente, en abrir todas las carpetas para descubrir su contenido y en abrir también algunos de los archivos con un editor de textos.

En total, encontramos más de 50 elementos, algunos de los cuales no sabíamos interpretar o identificar. Las extensiones de archivo eran numerosas, desde «.txt» hasta otras que no conocíamos, como «.glade», «.skip», «.am» o «.in».

Lo que sí intuimos es que la carpeta «po» era la que albergaba lo más importante —o, al menos, lo que sería nuestra prioridad como traductores—: las cadenas de la interfaz de usuario.

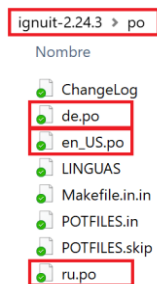


Ilustración 29. Archivos en la carpeta «po»

La extensión «.po» (*portable object*) nos llamó la atención, puesto que este tipo de archivo es el que a veces se usa para codificar programas o juegos. Efectivamente, dentro de la carpeta, encontramos archivos que parecían ser los correspondientes a la interfaz de usuario. Tim Musson había mencionado las traducciones de Ignuit al alemán y al ruso. Junto con el archivo «en\_US.po» estaban otros dos, nombrados como «de.po» y «ru.po», lo cual daba todavía más consistencia a nuestra teoría. Al abrir «en\_US.po» con un editor de textos, esto fue lo que apareció:

```
en_US.po
1 # US English translation of ignuit
2 #
3 # Copyright (C) 2009 Timothy Richard Musson
4 #
5 # This program is free software: you can redistribute it and/or modify
6 # it under the terms of the GNU General Public License as published by
7 # the Free Software Foundation, either version 3 of the License, or
8 # (at your option) any later version.
9 #
10 # This program is distributed in the hope that it will be useful,
11 # but WITHOUT ANY WARRANTY; without even the implied warranty of
12 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 # GNU General Public License for more details.
14 #
15 # You should have received a copy of the GNU General Public License
16 # along with this program. If not, see <http://www.gnu.org/licenses/>.
17 #
18 msgid ""
19 msgstr ""
20 "Project-Id-Version: ignuit\n"
21 "Report-Msgid-Bugs-To: \n"
22 "POT-Creation-Date: 2009-09-14 17:44+1200\n"
23 "PO-Revision-Date: 2009-09-14 17:56+1200\n"
24 "Last-Translator: Timothy Musson <trmusson@gmail.com>\n"
25 "Language-Team: <trmusson@gmail.com>\n"
26 "MIME-Version: 1.0\n"
27 "Content-Type: text/plain; charset=UTF-8\n"
28 "Content-Transfer-Encoding: 8bit\n"
29 "X-Poedit-Language: English\n"
30 "X-Poedit-Country: UNITED STATES\n"
31
32 #: ../data/glade/editor.glade.h:1
33 #: ../data/glade/main.glade.h:1
34 msgid "Add Card"
35 msgstr ""
36
37 #: ../data/glade/editor.glade.h:2
38 msgid "Add card"
39 msgstr ""
```

Ilustración 30. Vista del archivo «en\_US.po» de Ignuit en Notepad++

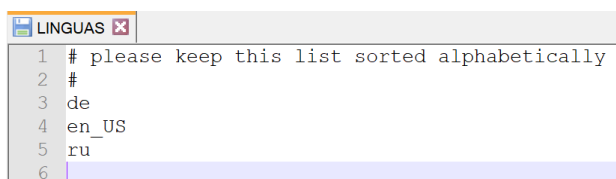
En la primera línea, vemos que pone «*US English translation of Ignuit*». Desconocíamos por qué decía que era una traducción —aunque más tarde descubrimos que era una adaptación al inglés de Estados Unidos hecha a partir del original en neozelandés—.

A partir de la línea 18, podemos identificar las cadenas de texto, junto con metadatos sobre la aplicación Ignuit (versión, nombre del desarrollador, codificación de caracteres, etc.).

El segmento en lengua de partida —o, en este caso, una posterior adaptación al inglés de EEUU—, se inicia con la cadena «msgid». El texto traducible va inmediatamente detrás, entrecomillado. Debajo encontramos la correspondiente traducción, indicada con «msgstr» y el texto traducible, también entrecomillado.

Con este sistema identificamos algunos archivos traducibles: «README», el archivo PO con la interfaz de usuario, y también algunos otros, como «ignuit.xml», «INSTALL», o

«COPYING». También nos dimos cuenta de que había otros datos que quizá tendríamos que modificar como tarea paralela a la traducción. El archivo «LINGUAS», ubicado en la carpeta «po», contenía una lista de las lenguas en las que estaba disponible Ignuit (alemán, inglés y ruso). Por otra parte, en el archivo «AUTHORS», aparecía no solo el nombre del desarrollador, sino también los nombres de los traductores de Ignuit al alemán y el ruso. En el propio «en\_US.po» hay una línea de metadatos para incluir los nombres de los traductores (ver «*Last translator*» en la ilustración 30 de la página anterior). Dimos por hecho que, ya fuéramos nosotros o bien Tim Musson, al terminar nuestras traducciones se tendrían que añadir nuestras lenguas al archivo «LINGUAS» y nuestros nombres a «AUTHORS» y el PO.



```

LINGUAS
1 # please keep this list sorted alphabetically
2 #
3 de
4 en_US
5 ru
6

```

Ilustración 31. Vista del archivo «LINGUAS» de Ignuit en Notepad++

Para poder llegar a detectar absolutamente todos los archivos traducibles de Ignuit, debíamos hacer muchas averiguaciones. Nos pareció más sencillo contactar de nuevo con el desarrollador para preguntarle directamente qué era lo localizable. De todas formas, independientemente de su respuesta, nos dimos cuenta de que si traducíamos el PO —y, si nos daba tiempo, quizá algo más—, sumando esto a la traducción de OmegaT, teníamos suficiente volumen de trabajo como para llegar a un mínimo de 2.000 palabras por persona. Por consiguiente, decidimos empezar por los archivos que sabíamos seguro que podíamos traducir y ver más adelante si podíamos hacer alguno más. Al final de este proyecto, los únicos archivos que modificamos fueron:

ES	CA	IT
en_US.po	en_US.po	en_US.po
ignuit.pot	ignuit.pot	ignuit.pot
README.txt	README.txt	README.txt
	ignuit.xml	

Ilustración 32. Archivos traducidos de Ignuit

Escogimos estos archivos porque pensamos que eran prioritarios. Un usuario que se descargue Ignuit verá la *GUI* con toda seguridad, pero además es muy probable que necesite consultar los archivos «README» e «ignuit.xml» en algún momento. Habría sido interesante traducir otros documentos, como «TODO», «INSTALL» o «ignuit.info», pero nuestro tiempo fue insuficiente.

Tim Musson nos confirmó más adelante que el PO correspondía a la interfaz de usuario, pero el archivo original no era aquel «en\_US.po» de la carpeta de descarga, sino otro archivo que nos envió por correo, el «ignuit.pot».

Los formatos PO y POT son ampliamente utilizados en localización. El formato POT resulta de extraer con el método *gettext*<sup>11</sup> las cadenas de texto traducibles de un programa a partir del código fuente inicial. La sintaxis sencilla de los archivos POT permite detectar

<sup>11</sup> *Gettext* es un paquete informático que facilita la internacionalización de *software*. Incluyendo funciones *gettext* en el código fuente de un programa se pueden extraer posteriormente determinadas cadenas de texto. Para más información, véase la referencia GNU Operating System (2016). *Gettext* [en línea] <https://www.gnu.org/software/gettext/>.


fácilmente las cadenas traducibles en lengua original. En un archivo de este tipo, las cadenas traducibles en lengua de partida se introducen con la expresión *msgid "xxx"*, donde «xxx» equivaldría al texto traducible. Esta expresión iría seguida de otra para las cadenas de texto traducidas: *msgstr ""*. Sin embargo, entre las comillas no encontraremos ningún texto; esta parte se deja en blanco y es en los archivos PO donde encontramos las traducciones.

Los archivos POT se usan como plantillas para originar archivos PO, los cuales contendrán las cadenas de texto tanto en lengua de partida como en lengua meta. Para cada lengua meta se crea un PO diferente. Así pues, para el caso que nos ocupa, de un único archivo POT («ignuit.pot») se generaron varios PO diferentes: tres ya existentes antes de nuestro trabajo («en\_US.po», «ru.po», «de.po») y tres que fueron producto de nuestra localización con OmegaT («ca.po», «it.po», «es.po»).

Al recibir el archivo «ignuit.pot» ya habíamos iniciado la traducción de «en\_US.po». El POT y el PO eran casi idénticos, pero el POT contenía algunas cadenas que no aparecían en el PO y que debían ser traducidas. Lo que hicimos entonces fue añadir el POT a nuestra carpeta «source». Con esta acción, pretendíamos que el POT se tradujera automáticamente a través de la autopropagación en OmegaT. Efectivamente, el proceso funcionó para el italiano y para el catalán. En cambio, por motivos que aún ahora desconocemos, no encontramos manera de conseguirlo para el español. Pensamos que esto se podía deber al código de país asociado a la lengua; sin embargo, nuestras comprobaciones nos demostraron que no era el motivo.

La solución fue crear un proyecto nuevo en OmegaT (EN>ES) y colocar el POT en la carpeta de archivos originales, «source». OmegaT permite pretraducir archivos a través de memorias de traducción. Solo es necesario colocar dichas memorias en la subcarpeta «tm\auto». Si en «tm\auto» colocábamos nuestro «project\_save.tmx» resultante de la traducción del PO al español, todos los segmentos que fueran iguales en el PO y el POT deberían quedar automáticamente traducidos. Lo probamos y así fue.

En cuanto a la compatibilidad de OmegaT con las extensiones de archivo con las que trabajábamos, no hubo grandes dificultades. Los archivos «.po», «.pot» y «.xml»<sup>12</sup> fueron reconocidos automáticamente por la aplicación; simplemente debíamos tener en cuenta —igual que pasaba con los archivos traducibles de OmegaT—, que algunas etiquetas de formato se tenían que mantener:



The image shows a snippet of XML code. The first line is `<b>Background Colours</b>` where the text "Background Colours" is highlighted in green. The second line is `<b>Colores de fondo</b><segment 0101>` where the text "Colores de fondo" is highlighted in blue.

Ilustración 33. Interpretación y visualización de las etiquetas de formato en OmegaT

El archivo «README» no fue reconocido al principio, pero bastó con añadirle manualmente la extensión «.txt» desde el explorador de archivos. Esta acción no dañó en absoluto el archivo y, en cambio, permitió que OmegaT lo identificara y lo mostrara correctamente.

### 3.3.2. Recuento de palabras

#### 3.3.2.1. OmegaT

El recuento de palabras de los *translatable*s de OmegaT lo realizamos con la misma herramienta OmegaT. La función «Herramientas > Estadísticas» extrae un recuento exhaustivo no solo de las palabras totales que conforman el texto, sino también del número de segmentos, del número de palabras y segmentos pendientes de traducir, del número de caracteres... En este apartado, aprovechamos para exponer también el número de palabras traducidas al entregar el proyecto.

<sup>12</sup> Usamos comillas latinas y minúsculas para referirnos a extensiones de archivo (ej.: «.xml»), pero nos referimos a los archivos «en\_US.po» e «ignuit.pot» como «el PO» y «el POT», respectivamente.

Las estadísticas resultantes de la primera extracción —antes de empezar a traducir— no fueron exactamente iguales para la combinación EN>ES que para la combinación EN>CA (por ejemplo, no apareció un mismo número en el total de palabras de los archivos originales). De igual manera, al sacar las estadísticas al final del proceso, el número de palabras totales del proyecto no fue igual al del recuento inicial dentro de una misma combinación lingüística. A pesar de esto, la operación sirve para hacerse una idea aproximada del número de palabras a traducir. Evidentemente, el número de palabras pendientes de traducción no era el mismo en el proyecto en español —mucho más avanzado— que para el catalán.

Sabemos gracias al «project\_save.tmx» que nadie ajeno a nuestro grupo editó el proyecto durante el tiempo que estuvimos trabajando en él, por lo que el número de palabras pendientes de traducir y traducidas se corresponde íntegramente con nuestra aportación.

Archivo(s)	N.º de palabras al empezar*		N.º de palabras traducidas	
	EN>ES	EN>CA	EN>ES	EN>CA
Todos (x73)	Total: <b>74.985</b> Pendientes: 47.799	Total: <b>75.097</b> Pendientes: 66.504	<b>2.928</b> (Todavía pendientes de traducción: 44.871)	<b>4.626</b> (Todavía pendientes de traducción: 61.878)
Resources.html	Total: 659 Pendientes: sin definir		Sin definir	
Spelling.html	Total: 489 Pendientes: sin definir		Sin definir	
Glossaries.xml	Total: 1.184 Pendientes: sin definir		Sin definir	
UserInterface.xml	Total: 3.869 Pendientes: 3.228		1.672	
Bundle.properties		Total: 7.991 Pendientes: 5.525		2.845
AboutOmegaT.xml		Total: 965 Pendientes: 965		854
InstantStartGuide.xml		Total: 759 Pendientes: 753		752
Corrección de cadenas sueltas traducidas por otras			Sin definir	175

personas anteriormente				
---------------------------	--	--	--	--

\*Nos basamos en las primeras estadísticas extraídas. El número total de palabras de cada archivo no resultó ser igual en el recuento final.

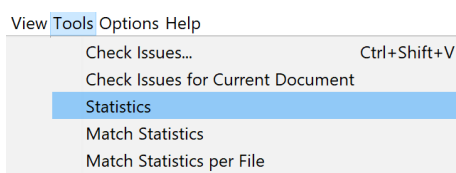


Ilustración 34. Interfaz de OmegaT. Función «Statistics»

Pensamos que una buena manera de hacerse una idea de las palabras traducidas es importar el recuento estadístico de OmegaT (se descarga en formato «.txt»), tanto antes de empezar como al finalizar, y luego usar una herramienta comparativa como, por ejemplo, [WinMerge](#), para ver la diferencia entre un documento y el otro. En nuestro caso, este recurso no fue del todo útil, ya que ni siquiera el número total de palabras del proyecto había dado el mismo resultado con OmegaT en las dos ocasiones. Vemos, de todas formas, una captura que muestra la comparación hecha por WinMerge:

Project Statistics				Project Statistics			
	Segments	Words	Characters (without spaces)		Segments	Words	Characters (without spaces)
Total:	8402	75005	404853	Total:	8406	75097	
Remaining:	6157	61878	331734	Remaining:	7007	66504	
Unique:	7731	72970	393288	Unique:	7736	73069	
Unique Remaining:	5817	60512	324407	Unique Remaining:	6575	64997	

Individual File Statistics:			Individual File Statistics:		
File Name	Total Segments	Remaining	File Name	Total Segments	Remaining
Bundle.properties	1859		Bundle.properties	1855	
doc_src/AboutOmegaT.xml	56		doc_src/AboutOmegaT.xml	54	
doc_src/App_Aknowledgements.xml	15		doc_src/App_Aknowledgements.xml	15	
doc_src/App_Keyboard.xml	60		doc_src/App_Keyboard.xml	60	
doc_src/App_Languages.xml	562		doc_src/App_Languages.xml	562	
doc_src/App_LanguageTool_plugin.xml	20		doc_src/App_LanguageTool_plugin.xml	20	
doc_src/App_LegalNotices.xml	22		doc_src/App_LegalNotices.xml	22	
doc_src/App_Scripting_plugin.xml	40		doc_src/App_Scripting_plugin.xml	40	
doc_src/App_ShortCutCustomization.xml	335		doc_src/App_ShortCutCustomization.xml	335	
doc_src/App_TeamProjects.xml	102		doc_src/App_TeamProjects.xml	102	
doc_src/App_Tokenizer_plugin.xml	23		doc_src/App_Tokenizer_plugin.xml	23	
doc_src/App_Website.xml	44		doc_src/App_Website.xml	44	
doc_src/Dictionaryes.xml	32		doc_src/Dictionaryes.xml	32	
doc_src/EditingBehavior.xml	68		doc_src/EditingBehavior.xml	68	
doc_src/FileFilters.xml	159		doc_src/FileFilters.xml	159	
doc_src/FilesAndFolders.xml	156		doc_src/FilesAndFolders.xml	156	
doc_src/FilesToTranslate.xml	146		doc_src/FilesToTranslate.xml	146	
doc_src/FormattedText.xml	119		doc_src/FormattedText.xml	119	

Ilustración 35. Interfaz de WinMerge. Comparación de estadísticas

### 3.3.2.2. Ignuit

El recuento para Ignuit lo hicimos con tres herramientas distintas, a fin de comparar el resultado obtenido con cada una.

Mostramos a continuación nuestro recuento inicial (capturas para la combinación EN>CA). Los archivos incluidos son: «en\_US.po», «README.txt» e «Ignuit.xml». Al no saber en ese momento que existía un «ignuit.pot», no pudimos incluirlo en el análisis.

-MemoQ:

## Statistics for file(s) [eng-cat]ignuit.xml, [eng-cat]en\_US.po, [eng-cat]README.txt

### Counts

Scope *Selected documents; number of documents: 3*

Type	Segments	Source words
All	914	4757
Repetition	176	291
X-translated / double context	0	0
Not started	889	4757
Pre-translated	25	0
Fragments	0	0
Edited	0	0
Rejected	0	0
Translator confirmed	0	0
Reviewer 1 confirmed	0	0
Reviewer 2 confirmed	0	0
Locked	0	0

Ilustración 36. Interfaz de MemoQ. Estadísticas de «en\_US.po», «ignuit.xml» y «README.txt»

-Matecat:

				
English US > Catalan 3292 MateCat Weighted words				
	Total	Weighted	New	Repetition
		Payable rate	100%	30%
File details (3)	4,902	3,292	0	80

Ilustración 37. Interfaz de Matecat. Estadísticas de «en\_US.po», «ignuit.xml» y «README.txt»

-OmegaT:

en-ca_first_report.txt			
1	31/10/18 17:16		
2	Project Statistics		
3			
4	spaces) #Files	Segments	Words
5	Total:	687	4917
6	28819	3	
6	Remaining:	682	4897
	28700	3	
7	Unique:	647	4751
	27843	3	
8	Unique Remaining:	642	4731
	27724	3	

Ilustración 38. Interfaz de OmegaT. Estadísticas de «en\_US.po», «ignuit.xml» y «README.txt»

Podemos observar que el recuento fue similar —entorno a las 4.000 palabras, 3 archivos en total—, aunque no idéntico.

Más tarde analizamos por separado el archivo «ignuit.pot» con las mismas herramientas.

Cabe recordar que el POT contenía muchas coincidencias totales o parciales con el PO, por lo que a pesar de sumar un total de unas 1.000 palabras, muy pocos segmentos eran nuevos. Interesa remarcar la casilla «*Unique words*», es decir, las palabras nuevas.

Sumando los 4 archivos, el total de palabras del proyecto según los resultados de OmegaT es de 5.963 (5.120 si contamos solo las nuevas del POT).

Mostramos a continuación el número de palabras traducidas (partiendo de un total de 5.963):

EN>ES	EN>CA	EN>IT
2.223	4.073	2.129



Tabla 7. Palabras traducidas de Ignuit al ES, CA e IT

-EN>ES:

Para calcular las palabras traducidas al español, al tener el archivo «ignuit.pot» en un proyecto de OmegaT a parte para autopropagar la traducción del PO, hicimos el cálculo observando las estadísticas de los dos proyectos, tal como sigue:

P(1) = proyecto 1: «en\_US.po» + «Ignuit.xml» + «README.txt»

P(2) = proyecto 2: «ignuit.pot»


a = N.º total palabras P(1 y 2) = 5.963

b = N.º palabras sin traducir P(1) = 3.678

c = N.º palabras sin traducir P(2) = 62

FÓRMULA:  $a - (b + c) = 5.963 - 3.740 = 2.223$

-EN>CA:


Statistics

Project Statistics

	Segments	Words	Characters (without spaces)	Characters (including spaces)	#Files
<b>Total:</b>	961	5,963	30,183		35,017
<b>Remaining:</b>	201	1,890	9,314		10,996
<b>Unique:</b>	695	4,984	25,275		29,481
<b>Unique Remaining:</b>	188	1,841	9,055		10,701
<					

Individual File Statistics:

File Name	Total Segments	Remaining Segments	Unique Segments	Unique Remaining Segments	Total Words	Remaining Words	Unique Words	Unique Remaining Words	#Files
en_US.po	244	11	238	11	892	20	856	20	
ignuit.pot	273	14	50	4	1,006	40	203	22	
ignuit.xml	381	176	346	173	3,656	1,830	3,522	1,799	
README.txt	63	0	61	0	409	0	403	0	

Ilustración 39. Interfaz de OmegaT. Estadísticas del proyecto de OmegaT (EN>CA) con archivo POT propagado

-EN>IT:

Statistics

Project Statistics

	Segments	Words	Characters (without spaces)	Characters (including spaces)	#Files
Total:	961	5,963	30,183		35,017
Remaining:	427	3,834	18,960		22,317
Unique:	695	4,984	25,275		29,481
Unique Remaining:	402	3,733	18,423		21,706
<					

Individual File Statistics:

File Name	Total Segments	Remaining Segments	Unique Segments	Unique Remaining Segments	Total Words	Remaining Words	Unique Words	Unique Remaining Words	#Files
en_US.po	244	6	238	6	892	8	856		8
README.txt	63	0	61	0	409	0	403		0
ignuit.xml	381	366	349	349	3,656	3,617	3,532		3,532
ignuit.pot	273	55	47	47	1,006	209	193		193

Ilustración 40. Interfaz de OmegaT. Estadísticas del proyecto de OmegaT (EN>IT) con archivo POT propagado

## 3.4. Preparación de materiales de referencia

### 3.4.1.1. OmegaT

Los repositorios existentes en GitHub para las combinaciones EN>ES y EN>CA ya incluyen en la carpeta «tm» una serie de memorias de traducción que facilitan el trabajo.

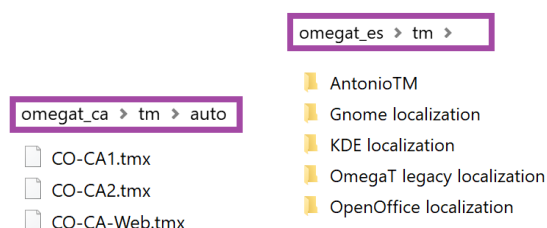


Ilustración 41. Memorias de traducción para el proyecto de OmegaT en la carpeta «auto» (EN>CA)

Los archivos «CO-CA1.tmx», «CO-CA2.tmx» y «CO-CA-Web.tmx» son las memorias de traducción procedentes de traducciones de versiones anteriores de OmegaT al catalán, incluyendo las de la página web. Están dentro de la subcarpeta «tm\auto», por lo que su contenido permite pretraducir las actualizaciones del *software*.

En la carpeta «tm» de la combinación EN>ES hay varias subcarpetas que contienen también memorias de traducciones hechas anteriormente.

Además de estas memorias, no se proporciona ningún otro material de referencia para OmegaT. Lo que hicimos fue usar los mismos glosarios y diccionarios que recopilamos para Ignuit y aplicarlos a OmegaT, colocando los archivos necesarios en las carpetas que les correspondía («glossary» y «dictionary»).

### 3.4.1.2. Ignuit

En este caso, partíamos totalmente de cero. La única referencia que teníamos eran las traducciones de la interfaz al alemán y el ruso. No obstante, nos pareció que algunas de las soluciones propuestas en dichas traducciones no eran del todo acertadas.

Identificamos varias tareas a realizar para obtener memorias y glosarios, y también diccionarios con los que configurar el corrector de OmegaT.

En un inicio, se nos ocurrió la idea de crear una MT alineando textos extraídos de páginas web cuyo ámbito fuera el mismo o similar al de Ignuit y con contenido disponible en nuestras lenguas. En otras palabras, queríamos buscar textos comparables. Encontramos varios sitios web desde los que se puede estudiar vocabulario, como las que nombramos a continuación:

- [Granule](#).
- [iMemorize](#).
- [Memorizar](#).
- [Quizlet](#).
- [Mnemosyne](#).
- [Pauker](#).

Curiosamente, las tarjetas mnemotécnicas de Ignuit se pueden convertir a formatos compatibles con algunas de estas aplicaciones.

Lo que pretendíamos hacer era descargar el código fuente de estos sitios web, tanto en inglés como en nuestras lenguas meta, y hacer una alineación usando una herramienta como, por ejemplo, MemoQ o LF Aligner. No obstante, al probarlo vimos en seguida que se trataba de una pretensión desmedida: en primer lugar, no todas estas páginas estaban disponibles en las lenguas que nos interesaban; en segundo lugar, para llegar a tener suficientes cadenas de texto necesitábamos seleccionar el contenido relevante de todo el sitio web; por último, el proceso de alineación habría llevado demasiado tiempo y no nos garantizaba un resultado eficaz. Además, el texto de nuestro TO no coincidía prácticamente en nada con el de los textos paralelos.

Para solventar este problema, acudimos al sitio web de corpus en línea [Opus](#) y nos descargamos varias memorias de traducción relacionadas con *software*: KDE, Ubuntu, GNOME, Android, Fedora y Debian. Intentamos conseguirlas en todas nuestras combinaciones. Las que no encontramos en Opus las descargamos de [Softcatalà](#). Al analizarlas detenidamente con el editor de textos Notepad++, detectamos incongruencias en algunas, aunque también cabe decir que nos ayudaron a comprender mejor el TO. Lo que decidimos fue aprovecharlas para nuestra traducción, pero no incluyéndolas en la carpeta «tm\auto» del proyecto, sino aplicándoles una penalización. Al añadir una memoria a la carpeta «tm\auto», se aprovechan los segmentos de dicha memoria que coinciden con los del proyecto en curso para hacer una traducción automática de las coincidencias exactas —es lo que se llama «pretraducción»—. En nuestro caso, no obstante, queríamos evitar precisamente esta pretraducción automática

porque las traducciones de las memorias de referencia no nos convencían del todo. Con el método de penalización podíamos aprovechar el contenido de estos materiales sin que se aplicaran automáticamente las traducciones a nuestro proyecto. Para incluir una MT penalizada a un proyecto de OmegaT, debemos tener dentro de la carpeta «tm» una subcarpeta llamada «penalty-xx», donde «xx» es un valor —generalmente, entre 10 y 40—, el cual asigna el usuario o usuaria manualmente y representa la penalización que se quiere aplicar a la MT sobre un valor total del 100%, que correspondería a una coincidencia exacta. En nuestro caso, aplicamos una penalización de valor 30, de modo que las coincidencias exactas se convirtieron en coincidencias parciales («fuzzy matches») del 70%. Tras la penalización, ningún segmento del proyecto presentaba una coincidencia del 100% con nuestras memorias de referencia y, por tanto, no se propagó ninguna traducción. En cambio, mantuvimos la ventaja de poder ver en el panel de edición la traducción de segmentos que seguían manteniendo un alto porcentaje de coincidencia.

Por otro lado, quisimos entrenar un motor de TA. Para ello, utilizaríamos MTradumàtica<sup>13</sup>. Sin embargo, dadas las indicaciones de nuestro profesor, desestimamos esta opción: para entrenar un motor TA era necesario contar con un corpus de millones de palabras, lo cual significaba que esta tarea también representaba en sí todo un proyecto.

Estimamos conveniente usar una base de datos terminológica para la traducción de Igñuit. Podíamos crear una de cero o ver si encontrábamos alguna que nos pudiera servir. Para crear una base de datos terminológica (o glosario), lo más apropiado era extraer términos de los archivos traducibles de Igñuit, determinar la traducción de los mismos en las tres lenguas meta, y después aplicarlas a la localización del programa.

Para extraer términos, solo disponíamos de SDL Multiterm en la universidad, y tampoco sabíamos muy bien cómo usarlo para este caso en concreto. Probamos entonces otras herramientas que habíamos visto en clase: [Vocab Grabber](#) y [File Filters](#). El problema fue que estos recursos son prácticos cuando la cantidad de archivos es menor y para otras extensiones de archivo. Lo que obtuvimos queda ilustrado en las imágenes siguientes:

Term	Occurrence	Word count
cards	10	1
i >Quiz menu	5	3
b >Unable	4	2
\ n POT-Creation-Date	1	3
b >Date	3	2
ignuit \n Report-Msgid-Bugs-To	1	3
ignuit flashcard	1	2
quiz window	1	2
statistics	2	1
check	2	1
Category	2	1
backs	2	1
fronts	2	1
card	1	1
choices	1	1

<sup>13</sup> Ver: Doğru, G.; Martín-Mor, A. (2017): *MTradumàtica: Free Statistical Machine Translation Customisation for Translators*.



Ilustración 43. Extracción de términos de Ignuit con Five Filters

Ilustración 44. Extracción de términos de Ignuit con Five Filters (2)

Ilustración 45. Extracción de términos de Ignuit con Vocab Grabber

La extracción de Five Filters fue algo más clara y precisa, aunque escasa. Decidimos, pues, que este método no era el más apropiado.

Lo que más podía ayudar era aprovechar bases de datos ya existentes relacionadas con la informática o hacer una extracción de términos manual. De hecho, combinamos las dos:

- Descargamos los archivos en formato «.tbx» correspondientes a la [terminología de Windows](#) (en EN, ES, IT y CA). El archivo correspondiente al inglés iría en la carpeta «glossary» de OmegaT en todas las combinaciones lingüísticas. El resto irían en la misma carpeta, cada uno para la lengua que le tocaba.

- Extrajimos de los archivos traducibles las palabras que serían consideradas términos y creamos una hoja de cálculo compartida en Google Drive con una columna para cada lengua que se fue completando poco a poco. La extracción la realizamos con AntConc. Aunque con AntConc es preferible utilizar archivos en formato «.txt», la aplicación los abrió sin problemas. Ejecutamos la función «*Word List*» y aislamos los términos de difícil comprensión.

Usamos el glosario propio como referencia externa; no lo añadimos a la carpeta «glossary» porque fuimos actualizándolo a medida que avanzábamos en la traducción.

Por otra parte, también como referencia externa, contamos con el glosario de Softcatalà, ahora recogido en la página «[Terminologia](#)».

Para encontrar la traducción de los términos, buscamos sobre todo en el servicio multilingüe [Cercaterm](#) de Termcat. Nos fue útil a la hora de buscar equivalencias en catalán y, aunque no siempre, para el español. En el caso del italiano empleamos diccionarios especializados de programación e informática y la terminología de Windows.

### 3.5. Selección de guías de estilo

Al principio, pensamos en diseñar nuestra propia guía de estilo para cada lengua a medida que traducíamos, pero tal propósito podía representar de por sí un proyecto para la asignatura de dimensiones considerables.

Pasamos, pues, a buscar alternativas existentes. Finalmente, optamos por basarnos en la guía de estilo de traducción del sistema operativo GNOME, disponible para nuestras tres combinaciones lingüísticas. La elección se basó, principalmente, en estos motivos:

- Uno de los programas que íbamos a localizar estaba desarrollado precisamente para su uso en GNOME.

- La guía de estilo de traducción de GNOME al catalán es la de Softcatalà, que es una de las principales referencias en localización en lengua catalana.

-GNOME es un sistema consolidado, y la guía para su traducción es completa, consistente y coherente.

-Podíamos usar la misma guía para las tres lenguas.

-Los puntos tratados en la guía de GNOME contenían mucha información relevante considerando las características de nuestro TO.

### 3.6. Instalación de un sistema operativo en un *Live USB* y de Ignuit

En primer lugar, tuvimos que instalar el sistema operativo Ubuntu en un *Live USB* y, a continuación, instalamos Ignuit. Para evitar que los cambios desaparecieran tras apagar el equipo, decidimos crear un *Live USB* persistente. Para ello contamos con la ayuda de nuestro compañero Marc. En primer lugar, era necesario saber con qué versión de Ubuntu podría funcionar Ignuit. A partir de las indicaciones del desarrollador y del documento «ignuit.xml» disponible en el paquete de descarga, requeríamos Ubuntu versión 16.04 Xenial (según el desarrollador, GNOME 2, que sería un entorno de escritorio de la distribución Ubuntu). Así pues, lo primero que hicimos fue descargar una imagen ISO de Ubuntu 16.04 Xenial y grabarla en un USB. Para ello utilizamos la aplicación Linux Live USB Creator 2.9.4. Una vez grabada en el USB, instalamos Ignuit. para ello fue necesario instalar desde el terminal de Ubuntu los paquetes indicados por el desarrollador en el documento «README».



Ilustración 46. Creación de un Live USB con Linux Live USB Creator

La instalación y uso de Ignuit han representado la porción más complicada de este proyecto. Varias veces intentamos instalar el sistema operativo persistente y varias veces probamos a instalar Ignuit sin éxito. En ocasiones sucedió también que al cerrar la sesión en el SO (sistema operativo) y volver a iniciar una sesión más tarde para seguir trabajando, la aplicación Ignuit aparecía en el sistema pero no se podía ejecutar, probablemente porque habíamos instalado mal alguno de los componentes y se había perdido en el cierre de sesión.

Solo en uno de nuestros USB conseguimos mantener Ubuntu e Ignuit para su uso, cosa que retrasó y dificultó significativamente nuestra labor.

### 3.7. Configuración de preferencias en OmegaT

Lo único que hacía falta una vez preparados los *translatable*s y los materiales de referencia era configurar OmegaT según nuestras preferencias. No explicaremos todas las opciones posibles, pero sí las que consideramos relevantes.

Una de las acciones requeridas fue marcar la opción «Allow translation to be equal to source» del menú «Opciones > Editor».

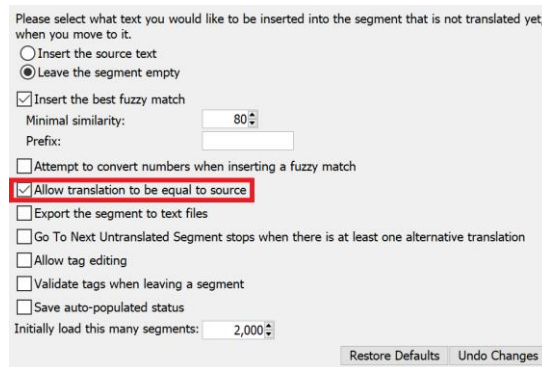


Ilustración 47. «Allow translation to be equal to source»

Para configurar el corrector, siguiendo las instrucciones del dossier de la asignatura OmegaT, descargamos el diccionario Hunspell de cada lengua meta y lo añadimos al proyecto a través de la carpeta «dictionary».

Cada miembro del grupo, en función de sus necesidades, configuró en OmegaT una serie de páginas para la búsqueda externa.

También nos aseguramos de que la función de autopropagación estaba activada.

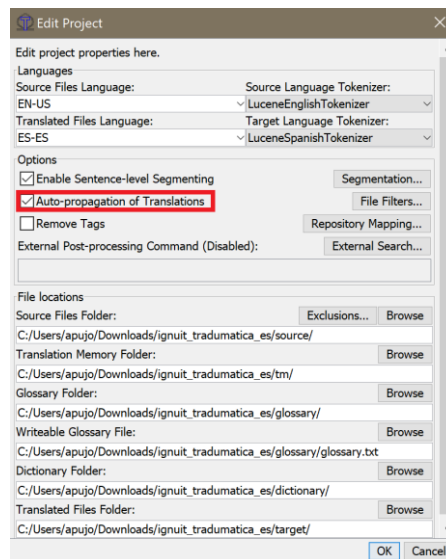


Ilustración 48. Casilla de verificación para la función de autopropagación en las preferencias de OmegaT

Cada traductor seleccionó una opción de entre las que se muestran en la siguiente captura, según lo que resultaba más cómodo a cada uno:

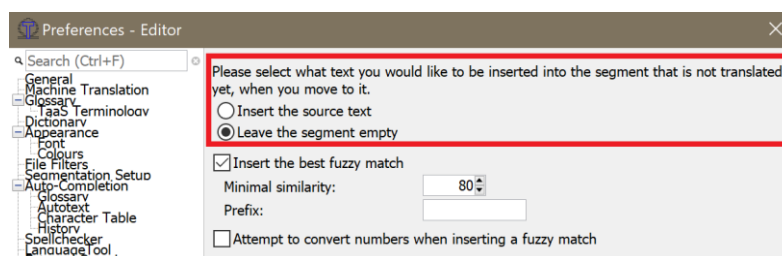


Ilustración 49. Preferencias en OmegaT. Comportamiento frente a segmentos no traducidos: insertar el texto original o dejar el segmento meta vacío

Un punto importante es la selección de motores de TA. Consideramos DeepL (traducción automática neuronal, o NMT), Google (TAE) y Apertium (TABR). Apertium lo habíamos probado en la asignatura *Fonaments de la localització* con OmegaT, obteniendo un grato resultado. Consideramos el traductor automático de Google puesto que es el que más

conocemos, y DeepL por consejo de un compañero ajeno al máster. Sin embargo, al observar que para obtener las API de Google y DeepL debíamos ingresar una mensualidad, descartamos estos motores.

Como indicamos en el apartado [2.3.1](#), al traducir, tuvimos en cuenta el usar siempre una misma ID de usuario. No siempre trabajamos desde el mismo ordenador, y esta acción fue útil para posteriormente filtrar nuestras traducciones y extraer una MT que contuviera solo los segmentos que habíamos editado nosotros.

En cuanto a las reglas de segmentación, mantuvimos las que OmegaT presentaba por defecto.

También configuramos el aspecto de la interfaz cada uno según sus propias preferencias (tamaño y tipo de fuente, colores, pestañas acopladas o desacopladas en el Editor, etc.).

Una función muy útil en OmegaT es la de «autocompletar», o «auto-text» en inglés. Se puede ajustar en «Opciones > Preferencias > Autocompletar».

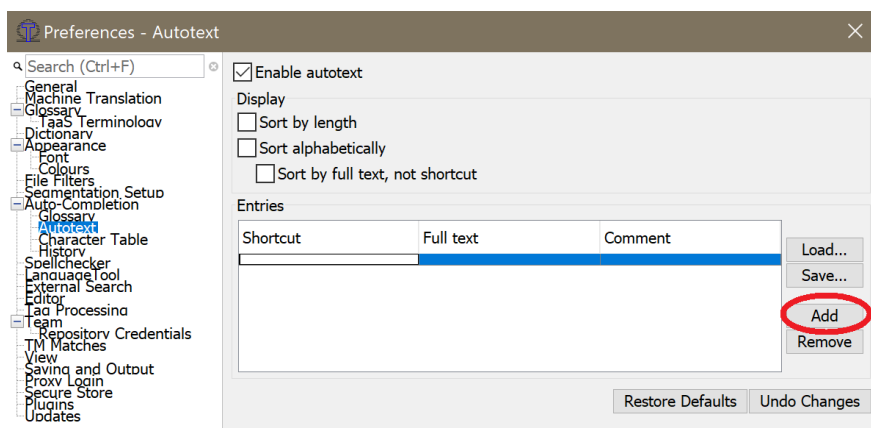


Ilustración 50. Función «auto-text» en las preferencias de OmegaT

De manera similar, el texto predictivo completa palabras automáticamente a medida que escribimos, extrayendo información de la memoria del proyecto para ello.

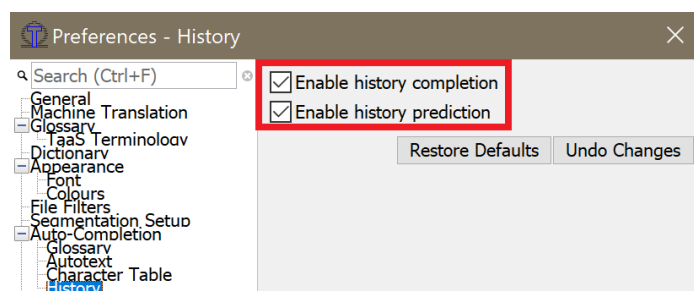


Ilustración 51. Preferencias de OmegaT. Habilidad del texto predictivo

Al tener que usar muchas comillas angulares, también denominadas «latinas», activamos la tabla de caracteres especiales y agregamos este tipo de comillas a la tabla, de modo que al pulsar Ctrl + Alt + Barra espaciadora OmegaT nos ofrecía la posibilidad de insertar caracteres de la tabla personalizada solo pulsando unas cuantas teclas.

## 4. Fase de traducción

### 4.1. Aspectos lingüísticos

El presente apartado no pretende servir como análisis traductológico exhaustivo sino como recolección de ejemplos de la problemática que puede aparecer durante el proceso de traducción de *software* y, más concretamente, del inglés al ES, CA e IT. Ilustramos algunos de



los problemas con los que nos encontramos, junto con las soluciones que adoptamos en cada caso.

Debido a que los problemas que identificamos eran similares para las dos aplicaciones localizadas, no haremos distinción entre OmegaT e Ignuit. La mayoría de los ejemplos expuestos a continuación son problemas comunes a las tres lenguas, aunque incluimos también algunos que son específicos de una combinación en particular.

### Estilo y tono

Muchas cadenas contenían oraciones en las que la máquina se dirigía al usuario de una manera muy directa. En nuestras tres lenguas meta, es más habitual despersonalizar las frases. Generalmente, la solución adoptada fue el cambio de sujeto. Veamos los siguientes ejemplos:

**You are going to modify {0} segments.**  
Aquesta acció modificarà {0} segments<segment 0498>

Ilustración 52. Ejemplo del cambio de sujeto en la oración (OmegaT, EN>CA)

**You'll also need:**  
Si necessita inoltre:<segmento 0650>

Ilustración 53. Ejemplo de despersonalización en la traducción EN>IT de OmegaT

Una de las cuestiones clave para nuestro trabajo fue el estilo del texto meta. En particular, nos centramos en la fórmula de tratamiento (tú, vos, usted...) y en el uso de infinitivo o imperativo para determinadas órdenes, funciones, botones, etc. En el apartado reservado a la fase de revisión (5.1), profundizamos más en este tema.

### Verbo elidido en el original

Es muy común en inglés encontrar frases en las que el verbo no aparece explícitamente, como en la captura. Nótese que en la traducción se ha construido una oración impersonal con el verbo «haber».

**No cards in Trash**  
No hay tarjetas en la Papelera<segment 0210>

Ilustración 54. Traducción al ES de «No cards in Trash» (Ignuit)

### Artículos y preposiciones

Generalmente, los segmentos originales no contenían partículas vacías de significado, como artículos y preposiciones —cosa que es muy frecuente en el inglés, y todavía más en los programas informáticos—. En muchas cadenas de texto añadimos estas partículas, puesto que en nuestras lenguas meta no es tan habitual su omisión.

**Google Api key not available.**  
La clau API de Google no està disponible.<segment 1138>

Ilustración 55. Inclusión de artículo en la traducción EN>CA de OmegaT

**Include m\_arkup in search**  
Includere la ma\_ratura nella ricerca<segmento 0035>

Ilustración 56. Inclusión de artículo en la traducción EN>IT de OmegaT

Esto es especialmente cierto para el catalán, sobre todo porque el uso de imperativo en lugar de infinitivo en muchos segmentos parecía demandar el uso de este tipo de palabras —porque la frase u oración resultaba mucho más natural—. Ej.: «*Comença el qüestionari*» contra «*Inizia questionario*» o «*Iniciar cuestionario*».



### Cambio de voz pasiva a voz activa

Esta diferencia en el uso de la lengua afecta, por lo general, a todos los tipos de texto cuando se traduce del inglés a una lengua latina. Durante el proceso de traducción, encontramos ejemplos de ello. En la captura, «...has been blocked» se sustituye por «...està bloquejada» («está bloqueada»):

**This API key has been blocked**  
Aquesta clau API està bloquejada<segment 1157>

Ilustración 57. Inclusión de artículo en la traducción EN>CA de OmegaT

Otro ejemplo es la traducción de «*Copying and distribution of this file, with or without modification, are permitted provided the copyright notice and this notice are preserved*» por «Se permite la copia y distribución de este archivo, modificado o no, siempre y cuando el aviso de copyright y el presente aviso se preserven» (Ignuit).

### Vocabulario especializado

Nuestra inexperiencia en lo que atañe a *software* provocó que tuviéramos que buscar información sobre determinados elementos que desconocíamos y para los que, en consecuencia, no encontramos intuitivamente un equivalente. Nótese a partir de la siguiente imagen que estos casos dieron uso a la función de búsqueda externa que integra OmegaT (buscamos el término «*pop directional formatting*» directamente en la Wikipedia sin tener que abrir manualmente la página en el navegador):

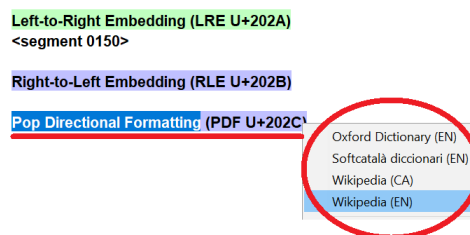


Ilustración 58. Vista del uso de la función de búsqueda externa en OmegaT

### Diferencias en el uso de la puntuación

Este es también un problema de traducción muy recurrente en todos los géneros y en la mayoría de combinaciones lingüísticas. Veamos cómo se resolvieron algunos casos en nuestra tarea de localización:

**Unable to convert project!**  
No s'ha pogut convertir el projecte<segment 0423>

Ilustración 59. Omisión de signo de exclamación en la traducción EN>CA de OmegaT

**If there's anything**  
**you're interested in, get in touch!**  
Se c'è qualcosa  
a cui sei interessato, mettiti in contatto.<segmento 0677>

Ilustración 60. Omisión de signo de exclamación en la traducción EN>IT de Ignuit

**OmegaT uses glossaries containing translations of single words or small phrases: a simplified bilingual dictionary for a specific domain.**  
L'OmegaT fa servir glossaris que contenen traduccions de paraules soltes o expressions curtes; és a dir, diccionaris bilingües senzills per a temàtiques específiques.<segment 1873>

Ilustración 61. Cambio en la estructura de la oración en la traducción EN>CA de OmegaT

También ajustamos algunos segmentos para conseguir un efecto estético más agradable.

**(Leave the Password empty if you are using an API Key.)**

(Deixeu el camp Contrasenya buit si feu servir una clau API)<segment 1149>

Ilustración 62. Elisión de signos de puntuación excesivos en la traducción EN>CA de OmegaT

### Traducción vacía de «please»

Es sabido entre los localizadores que la palabra «*please*» suele desaparecer cuando se traduce del inglés a varias otras lenguas —entre las cuales, las de este proyecto—.

**Please select the dictionaries you want to install and press the Install button.**

Seleccioneu els diccionaris que voleu instal·lar i premeu el botó Instal·la.<segment 1080>

Ilustración 63. Traducción vacía de «*plase*» en la traducción EN>CA de OmegaT

**Please check that this is an ignuit flashcard file.**

Verifique que este es un archivo de tarjetas de Ignuit.<segment 0239>

Ilustración 64. Traducción vacía de «*plase*» en la traducción EN>ES de Ignuit

**Please check that imported file is not compressed, and that the correct filter is selected.**

Verifica che il documento importato non sia un file compresso e che il filtro corretto sia selezionato.<segmento 0243>

Ilustración 65. Traducción vacía de «*plase*» en la traducción EN>IT de Ignuit

### Posesivos

En la traducción EN>ES/IT/CA, es muy frecuente la sustitución de determinantes posesivos por otros tipos de determinante:

**Your project does not contain any files in a supported file format.**

Aquest projecte no conté cap fitxer que tingui un format compatible.<segment 0077>

Ilustración 66. Uso de determinante demostrativo en la traducción EN>CA de OmegaT

**If your source files are formatted files, e.g. Microsoft Word, LibreOffice Writer or HTML, OmegaT will convert the formatting into tags that surround the text that you translate.**

Si els fitxers de partida són fitxers amb for<segment 4276>

Ilustración 67. Uso de artículo definido en la traducción EN>CA de OmegaT

**<e0> Translation memory:</e0><a1> OmegaT</a1> stores your translations in a translation memory file.**

<e0>Memòria de traducció:</e0><a1>L'OmegaT</a1> desa les traduccions en un fitxer amb una memòria de traducció.<segment 1863>

Ilustración 68. Uso de artículo definido en la traducción EN>CA de OmegaT

### Cuestiones de convenciones formales o de concepto

En inglés, los títulos en general se escriben usando mayúsculas para las primeras letras de las palabras principales, cosa que no suele hacerse en ninguna de nuestras lenguas meta. Esto nos lleva a adaptar el texto meta, como ocurre en el ejemplo siguiente, en el que el estilo «*title*» («título») se sustituye por «*títol anglès*» (es decir, estilo «título inglés»):

Canvia el tipus de lletra a

M&inúscula

M&ajúscula

Estil &títol anglès

Estil &oració

Ilustración 69. Diferencias conceptuales EN>CA extraídas de la traducción EN>CA de OmegaT

### Uso de mayúsculas y minúsculas

En inglés, es común el uso de las mayúsculas aun cuando no se trata del inicio de una frase u oración, mientras que en nuestras tres lenguas meta no es habitual.

Long Term Memorization

Memorización a largo plazo<segment 0571>

Ilustración 70. Uso de mayúsculas y minúsculas en la traducción EN>ES de Ignuit

Reset to Default Schedules

Ripristina pianificazioni predefinite<segmento 0185>

Ilustración 71. Uso de mayúsculas y minúsculas en la traducción EN>IT de Ignuit

### Trasposición

En determinadas oraciones, hizo falta cambiar la estructura de la oración para mantener la naturalidad en el orden del tema-remata en la lengua meta.

Only cards marked as <i>Known</i> or <i>Unknown</i> are recorded.

Vengono registrate solo le schede contrassegnate come <i>Appresa</i> o <i>Non appresa</i>. <segmento 0235>

Ilustración 72. Trasposición en la traducción EN>IT de OmegaT

### Uso de «per» y «per a» delante de infinitivo (CA)

Softcatalà aboga por el uso de «per» delante de infinitivo, tomando como referencia la norma de Coromines y Solà (ver [Bibliografía](#)). Por otra parte, nos fijamos en que en todas las versiones anteriores de OmegaT se había seguido esta regla. Así pues, decidimos aplicarla también a nuestras traducciones, aunque algunos de nosotros somos partidarios de la norma de P. Fabra.

### Nuevos diacríticos (CA)

En catalán, debemos recordar la nueva norma con respecto a los acentos diacríticos (IEC, 2016<sup>14</sup>) y tener en cuenta que, por defecto, OmegaT sigue configurado según la norma antigua.

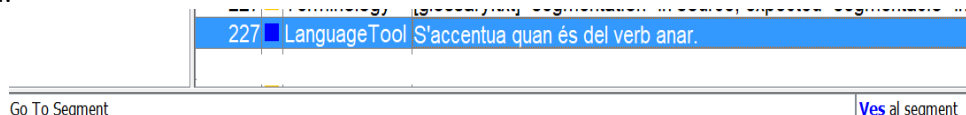


Ilustración 73. Acentos diacríticos. Falso positivo en el QA con OmegaT: «ves», del verbo «ir» en catalán

La detección de falsos errores se puede evitar desde «Preferencias > LanguageTool»:

<sup>14</sup> Ver el siguiente enlace a ÉsAdir: <http://esadir.cat/gramatica/criteris/diacritics>

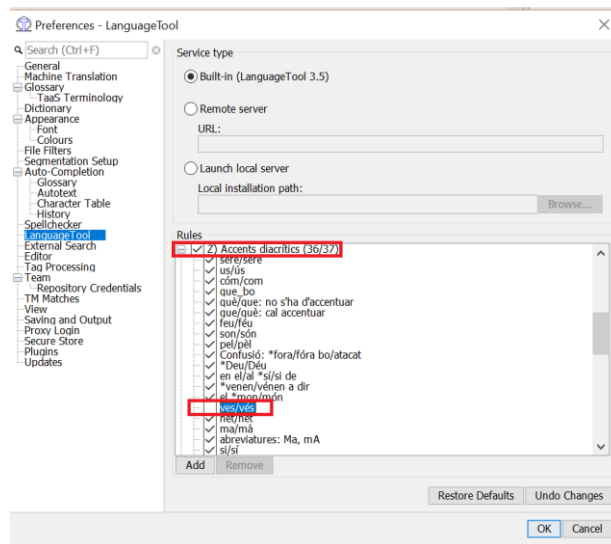


Ilustración 74. Preferencias de corrección ortográfica del LanguageTool de OmegaT: dejar de corregir el diacrítico en «ves» en catalán

## 4.2. Aspectos de formato y localización

En este apartado se podrían incluir también algunos de los ejemplos mencionados arriba, como el hecho de que en *software* y, más concretamente, en inglés, las frases son más sencillas. Detallamos a continuación algunos casos más.

### Falta de contexto

Uno de los mayores problemas con los que podemos toparnos al localizar *software* es la falta de contexto; OmegaT e Ignuit no han sido una excepción: incluso teniendo acceso al programa, nos encontramos con cadenas de texto que no sabíamos situar en la interfaz. Por otro lado, es importante remarcar que no llegamos a ser capaces de instalar o utilizar Ignuit hasta una fase muy avanzada de nuestro proyecto; de hecho, no fue hasta el momento de la revisión y el testeo.

Se incluyen en esta categoría todos aquellos segmentos cortos y aislados que podían dar lugar a dudas.

En la siguiente imagen vemos un ejemplo de falta de contexto y un segmento de una sola palabra, ambos de Ignuit. En cuanto al primero de estos problemas, nos referimos al término «*sticky flips*», funcionalidad de Ignuit que sirve para evitar que las tarjetas se giren automáticamente al pasar de una a otra —por defecto, si estamos viendo el reverso de una tarjeta, al pasar a otra se muestra el anverso de esta, pero con «*sticky flips*» activado, seguiríamos viendo solo reversos—. Para entender este concepto tuvimos que consultar a Tim Musson.

La otra cuestión que se aprecia en la captura es la duda entre si «*Tag*» actúa como nombre o como verbo. Intuimos que se trataba de un verbo, pues la cadena se encontraba dentro de una serie de segmentos en los que se indicaban acciones. Aun así, no pudimos comprobarlo hasta el momento del testeo.

### Sticky Flips

Cambiar de cara

### Tag

Etiquetar|<segment 0020>

Ver detalles de la tarjeta

Ver el reverso de la tarjeta

Ilustración 75. Término «*sticky flips*» sin traducir y segmento «*Tag*» interpretado como verbo en la traducción EN>ES de Ignuit

Nos causaron también una serie de dudas los segmentos «*Listen to Fronts*», «*Listen to Backs*» y similares, porque al principio habíamos interpretado el «*to*» como direccional («hacia» algún sitio), y no como parte de un verbo que gramaticalmente demanda la preposición para expresar su significado («escuchar» + complemento directo).

### Listen to Front Automatically

Riprodurre parte frontale automaticamente<segmento 0134>

Ilustración 76. Segmento «*Listen to Front Automatically*» de Ignuit traducido al IT

Por último, otro ejemplo de la importancia del contexto es el segmento «*Back*». Cuando esta palabra aparecía suelta, no sabíamos si hacía referencia al reverso de una tarjeta o a la acción «*go back*».

### Consenso de traducciones

Por el hecho de traducir simultáneamente a tres lenguas, hubo que tomar algunas decisiones que garantizaran la cohesión entre versiones. Dos claros ejemplos de ello son la elección de las palabras «tarjeta» (ES), «*targeta*» (CA) y «*scheda*» (IT) para referirnos a «*card*» (original en EN).

Nos costó también establecer las traducciones de «*Front*» y «*Back*» en Ignuit para referirnos a la parte de delante y la de atrás de las tarjetas electrónicas. En español y catalán, teníamos una opción muy viable con «anversos» y «reversos», pero en italiano resultó que era muy difícil expresar esta idea con una sola palabra. Nuestra solución fue «*frontale*» y «*retro*» en la mayoría de casos en los que los términos aparecían sueltos, y «*parte frontale*» / «*parte retro*», con sus respectivos plurales, en la mayoría del resto de cadenas.

### Especificidades del género textual

Los ejemplos más claros son: las *hotkeys*, la identificación del texto traducible, las etiquetas, las palabras o frases cortas que funcionan perfectamente en el original pero que son difíciles de adaptar en la traducción y la falta de espacio para segmentos traducidos que son más largos que los originales.

Las *hotkeys* se introducen con «&» en OmegaT y con «\_» en Ignuit. Es importante determinar correctamente la letra que irá ligada a una *hotkey* para que la función tenga utilidad a la hora de usar la aplicación. A veces, la letra escogida no es la misma en el original que en la traducción, ni la misma tampoco en todas las traducciones. La mejor forma de comprobar que se ha escogido una letra adecuada es a través del testeo.

Para las *hotkeys*, intentamos elegir letras mayúsculas, que no fueran muy cortas (como la «i» o la «l» y que no se sobresalieran por debajo de la base de los caracteres (como la «g» o la «j»). Siempre que fue posible, mantuvimos la misma letra que en inglés, y siempre que identificamos que dos segmentos con *hotkeys* aparecían en el mismo menú, panel o ventana, evitamos el uso de una misma letra en *hotkeys* de palabras distintas.

&OK  
&Aceptar<segment 0039>  
&Cancelar

Ilustración 77. Presencia de *hotkeys* en la traducción de OmegaT

\_Cuestionario  
Barra de e\_estado  
\_View  
gtk-acerca-de  
gtk-buscar  
gtk-nuevo  
gtk-abrir  
gtk-preferencias

Ilustración 78. *Hotkeys* y marcador «gtk» en segmentos de Ignuit

La captura de arriba muestra, además, cadenas que en un principio no supimos interpretar: las que empiezan por «gtk». Más tarde, el desarrollador nos informó de que la expresión «gtk» sirve para «llamar» a las cadenas en las que aparece el texto que va detrás del «gkt». La función está vinculada a la aplicación [GTK+](#) para diseñar interfaces de usuario gráficas. Por ser órdenes que la máquina debe interpretar y que probablemente se leen directamente en el terminal, pensamos que era mejor no utilizar caracteres acentuados en estas cadenas.

gtk-about  
gtk-informazioni su <segmento 0090>

Ilustración 79. Ejemplo de cadena con marcador «gtk» en la traducción EN>IT de Ignuit

### Referencias a otras partes del programa o a materiales de documentación

Son muy frecuentes en localización las referencias a otras partes de la interfaz o a los materiales de documentación. Es especialmente difícil mantener siempre los mismos nombres en todas las referencias y que estos coincidan con la traducción del elemento referente —en especial cuando son muchas las personas que intervienen en el proceso—.

See OmegaT manual for instructions.

Consulte las instrucciones en el Manual de OmegaT.<segment 1143>

Ilustración 80. Referencia a un archivo de la documentación de OmegaT en cadenas de la interfaz de usuario (EN>ES)

En la imagen de abajo, vemos la alusión al archivo «TODO» de la documentación de Ignuit en la traducción EN>ES. Puesto que el mencionado archivo nunca ha sido traducido al español y el título del mismo sigue siendo «TODO», así lo mantuvimos en la traducción del segmento de la imagen.

See the TODO file.

Consulta el archivo «TODO».<segment 0948>

Ilustración 81. Referencia a un archivo de la documentación de Ignuit en cadenas del archivo «README.txt» (EN>ES)

Durante la fase de revisión, leímos en la [Guia d'estil de Softcatalà](#) que en estos casos lo que se recomienda es escribir el nombre original del referente y añadir una posible traducción a la lengua meta entre paréntesis. Esta técnica se podría haber usado perfectamente en nuestra traducción al español. Nos encontramos ante una de las posibles mejoras a realizar en nuestro trabajo. Sin embargo, cuando detectamos esta cuestión, nuestros esfuerzos estaban puestos en la redacción de la presente memoria, y no nos quedó tiempo de revisar y corregir todos los casos en los que ocurría algo parecido a lo que muestra la captura. Asimismo, debemos ajustar el uso de comillas, tanto para alusiones a la documentación como a botones o menús de la *GUI* en ambos programas.

## 5. Fase de postproducción

### 5.1. Revisión

#### 5.1.1. Revisión individual

Una vez finalizado el trabajo de traducción, cada uno revisó su propio trabajo antes de enviarlo al revisor correspondiente. Además de una lectura minuciosa de la propia traducción, ejecutamos el control de calidad de OmegaT sobre los archivos traducidos.

Recordemos que previamente habíamos configurado el diccionario Hunspell y las opciones de corrección.

En OmegaT, la función de control de calidad (*QA, quality assurance*) se denomina «Comprobar problemas» y está bajo el menú «Herramientas».

La herramienta pide que se seleccione el tipo de «problema» a comprobar. El usuario debe marcar las opciones que le interese incluir en el *QA* de entre: etiquetas, ortografía, terminología y LanguageTool. El [LanguageTool](#) es un complemento (o *plug-in*) bastante completo que detecta errores varios, como, por ejemplo, de estilo o gramaticales.

Una vez hecha la selección, OmegaT muestra una lista de los errores encontrados, divididos por tipología: todos, LanguageTool, etiquetas y terminología.

El primer *QA* con OmegaT resultó en un altísimo número de errores. Veámoslo con capturas de pantalla:

Options	Segment	Type	Description
All (711)			
LanguageTool (26)			
Terminology (685)			
	5986	terminology	MicrosoftTermCollection.tbx, EuroVoc ENES.tbx, container in source; expected one of "con
	5986	terminology	MicrosoftTermCollection.tbx, menu bar in source; expected barra de menús in target
	5986	terminology	MicrosoftTermCollection.tbx, severity in source; expected gravedad in target
	5986	terminology	MicrosoftTermCollection.tbx, availability in source; expected disponibilidad in target
	5987	terminology	MicrosoftTermCollection.tbx, protection in source; expected proteccion in target
	5987	terminology	MicrosoftTermCollection.tbx, setting in source; expected valor in target
	5987	terminology	MicrosoftTermCollection.tbx, set in source; expected conjunto in target
	5987	terminology	MicrosoftTermCollection.tbx, translate in source; expected trasladar in target
	5987	terminology	MicrosoftTermCollection.tbx, edition in source; expected edición in target
	5987	terminology	MicrosoftTermCollection.tbx, Edit in source; expected Edición in target
	5987	terminology	MicrosoftTermCollection.tbx, edit in source; expected one of edición, editar, modificar
	5987	terminology	MicrosoftTermCollection.tbx, type in source; expected one of escribir, tipo in target
	5987	LanguageTool	Esta frase no se inicia con mayúscula
	5987	terminology	MicrosoftTermCollection.tbx, sedimentation in source; expected sedimentación in target
	5987	terminology	MicrosoftTermCollection.tbx, translate in source; expected trasladar in target
	5987	terminology	EuroVoc ENES.tbx, MicrosoftTermCollection.tbx, display in source; expected one of "anunc

Ilustración 82. Número total de errores en el primer *QA* de la traducción EN>ES de OmegaT

Opzioni	Segmento	Tipo	Descrizione
Tutto (540)			
LanguageTool (44)			
Terminologia (496)			
	20	Terminologia	MicrosoftTermCollection.it.tbx, tag nella fonte; attesa una di "aggiungere un tag", "contrassegnare", "contrassegno", "tag" nella destinazione
	21	Terminologia	MicrosoftTermCollection.it.tbx, Detail view nella fonte; attesa "visualizzazione Dettagli" nella destinazione
	21	Terminologia	MicrosoftTermCollection.it.tbx, Viewed nella fonte; attesa "Visualizzati" nella destinazione
	21	Terminologia	MicrosoftTermCollection.it.tbx, view nella fonte; attesa una di "vista", "visualizzare", "visualizzazione" nella destinazione
	22	Terminologia	MicrosoftTermCollection.it.tbx, Viewed nella fonte; attesa "Visualizzati" nella destinazione
	22	Terminologia	MicrosoftTermCollection.it.tbx, view nella fonte; attesa una di "vista", "visualizzare", "visualizzazione" nella destinazione
	22	Terminologia	MicrosoftTermCollection.it.tbx, closing nella fonte; attesa "chiusura" nella destinazione
	25	Terminologia	MicrosoftTermCollection.it.tbx, closed nella fonte; attesa una di "chiusa", "chiuso" nella destinazione
	25	Terminologia	MicrosoftTermCollection.it.tbx, dose nella fonte; attesa una di "chiudere", "chiusura" nella destinazione
	25	LanguageTool	Questa frase non inizia con una maiuscola
	26	LanguageTool	Questa frase non inizia con una maiuscola
	27	Terminologia	MicrosoftTermCollection.it.tbx, cut nella fonte; attesa "tagliare" nella destinazione
	27	LanguageTool	Questa frase non inizia con una maiuscola
	27	Terminologia	MicrosoftTermCollection.it.tbx, delete nella fonte; attesa "eliminare" nella destinazione
	28	LanguageTool	Questa frase non inizia con una maiuscola
	29	Terminologia	MicrosoftTermCollection.it.tbx, paste nella fonte; attesa "incollare" nella destinazione
	29	Terminologia	MicrosoftTermCollection.it.tbx, Past nella fonte; attesa una di "chiusa", "chiuso" nella destinazione
	29	LanguageTool	Questa frase non inizia con una maiuscola
	30	Terminologia	MicrosoftTermCollection.it.tbx, redo nella fonte; attesa una di "fase di rollover", "rollover" nella destinazione
	30	LanguageTool	Questa frase non inizia con una maiuscola
	31	Terminologia	MicrosoftTermCollection.it.tbx, undo nella fonte; attesa una di "annullare", "fase di rollback" nella destinazione
	31	LanguageTool	Questa frase non inizia con una maiuscola
	34	Terminologia	MicrosoftTermCollection.it.tbx, find nella fonte; attesa "trovare" nella destinazione
	35	Terminologia	MicrosoftTermCollection.it.tbx, regular nella fonte; attesa "normale" nella destinazione
	38	Terminologia	MicrosoftTermCollection.it.tbx, case nella fonte; attesa una di "case", "caso", "cassa", "maiuscole/minuscole" nella destinazione

Ilustración 83. Número de errores en el primer *QA* del archivo «en\_US.po» en la traducción EN>IT de Ignuit



Issues (10,787)			
Options			
All (10,787)	Segment	Type	Description
LanguageTool (760)	7	terminology	MicrosoftTermCollection.tbx   "projection" in source: expected "proyección" in target
Tags (86)	7	terminology	MicrosoftTermCollection.tbx   "optional" in source: expected "opcional" in target
Terminology (9,941)	7	terminology	MicrosoftTermCollection.tbx   "usage" in source: expected "utilización" in target
	8	terminology	MicrosoftTermCollection.tbx   "projection" in source: expected "proyección" in target
	8	terminology	MicrosoftTermCollection.tbx   "specifier" in source: expected "especificador" in target
	8	terminology	MicrosoftTermCollection.tbx   "optional" in source: expected "opcional" in target
	8	terminology	MicrosoftTermCollection.tbx   "launch" in source: expected one of "iniciar", "inicio" in target
	8	terminology	MicrosoftTermCollection.tbx   "team" in source: expected "Equipo" in target
	9	terminology	glossary.txt   MicrosoftTermCollection.tbx   "team" in source: expected one of "trabajo en"
	9	terminology	MicrosoftTermCollection.tbx   "team" in source: expected "Equipo" in target
	10	terminology	MicrosoftTermCollection.tbx   "usage" in source: expected "utilización" in target
	11	terminology	EuroVoc ENES.tbx   "download" in source: expected "telecarga" in target

Ilustración 84. Número total de errores en la traducción EN>ES de OmegaT

A priori, puede parecer que la persona encargada de la traducción ha cometido un sinnúmero de fallos. Si nos fijamos mejor, muchos de los errores son debidos a variaciones morfológicas o lematizaciones de palabras del glosario, a frases que no empiezan con mayúscula —porque no deberían— o a signos de puntuación supuestamente mal utilizados.

Según creemos, hay varias maneras de enfrentarse a este problema:

1. Eliminar el glosario de la carpeta correspondiente del proyecto (poco aconsejable, pues nos interesa mantenerlo).
2. Ignorar los errores considerados falsos positivos (a riesgo de descuidarse y no detectar alguno que sí es relevante).
3. Intentar configurar las opciones de corrección para que determinados fallos no se detecten como tal (puede ser peligroso si para algunos segmentos un mismo caso se considera un error pero en otros segmentos, no).

La cuestión de los errores de terminología supuso una molestia para nosotros. Optamos por la solución número 2.

En esta primera revisión, la comunicación entre los integrantes del grupo fue muy importante porque nos ayudó a acordar entre todos algunos términos o a tomar ciertas decisiones. La herramienta Skype fue también muy útil para este fin.

Durante la traducción y la revisión individual de OmegaT, nos sirvió de apoyo la memoria de traducción «project\_save.tmx» disponible en GitHub. Es importante destacar que el «project\_save.tmx» es un historial de cambios y no la memoria de traducción definitiva. Las memorias definitivas se guardan como «xxx-level1.tmx», «xxx-level2.tmx» y «xxx-omegat.tmx» tras la generación de los materiales traducidos. En el «project\_save.tmx» se registran todos los cambios realizados durante el proceso de traducción y, en consecuencia, este documento podría mostrar algunas cadenas con traducciones que en algún momento se han desestimado. Toda persona que quiera consultar el «project\_save.tmx» como referencia deberá tener en cuenta este hecho. Aun así, puede ser un gran apoyo, especialmente cuando se quiere comprender el sentido del original y se observan traducciones a otros idiomas. Por ejemplo, al traducir el archivo «Bundle.properties» al catalán, pudimos aprovechar el «project\_save.tmx» de la traducción al español como muleta.

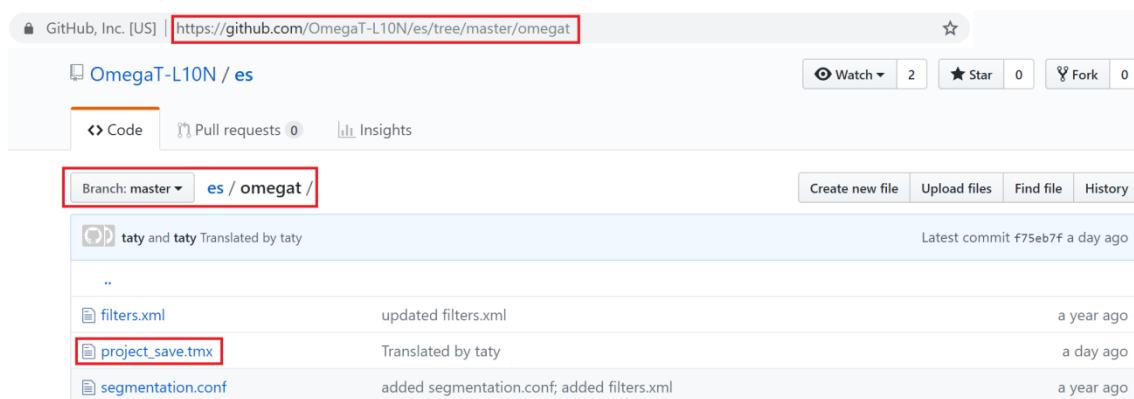


Ilustración 85. Archivo «project\_save.tmx» de la traducción EN>ES de OmegaT en el repositorio GitHub



```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tmx SYSTEM "tmx11.dtd">
<tmx version="1.1">
  <header creationtool="OmegaT" o-tmf="OmegaT TMX" adminlang="EN-US" datatype="plaintext" creationid="Cristina" creationdate="20170916T171645Z" />
  <!-- Default translations -->
  <tu>
    <tuv lang="EN">
      <seg>"Aligner" aligner utility</seg>
    </tuv>
    <tuv lang="ES" changeid="Cristina" changedate="20170916T171645Z" creationid="Cristina" creationdate="20170916T171645Z">
      <seg>Utilidad de alineación Aligner</seg>
    </tuv>
  </tu>
  <tu>
    <tuv lang="EN">
      <seg>"Bligner" aligner utility</seg>
    </tuv>
    <tuv lang="ES" changeid="Cristina" changedate="20170916T171811Z" creationid="Cristina" creationdate="20170916T171811Z">
      <seg>Utilidad de alineación Bligner</seg>
    </tuv>
  </tu>
</tmx>

```

Ilustración 86. Muestra de una consulta al «project\_save.tmx» en GitHub de la traducción EN>ES de OmegaT

Como ejemplo, pongamos por caso que queremos buscar cómo se ha traducido al español el segmento «S&space matches nbsp».

Segment Properties		Editor - Bundle.properties
Name	Value	modificat a&bans del:
File	Bundle.properties	Modificat a&bans del:
ID	SW_SEA RCH_SP ACE_MA TCH_NB SP	Distingeix entre ma&júscules i minúscules
Has note	true	No distingeix entre &caràcters d'amplada completa i mitja
Changed on	Nov 15, 2018	S&space matches nbsp
		Es&pais no separables<segment 0477>
		Cerca a

Ilustración 87. Segmento «S&space matches nbsp» en la traducción EN>CA de OmegaT.

En el navegador web, con el «Project\_save.tmx» EN>ES abierto, buscamos el texto objeto de duda.

```

</tu>
<tu>
  <tuv lang="EN">
    <seg>S&space matches nbsp</seg>
  </tuv>
  <tuv lang="ES" changeid="hcartagena" changedate="20151209T181634Z" creationid="hcartagena" creationdate="20151117T192832Z">
    <seg>Incluir es&space;pacios duros</seg>
  </tuv>
</tu>

```

Ilustración 88. Búsqueda con Ctrl+F de la expresión «nbsp» en el «project\_save.tmx» de la traducción EN>ES de OmegaT en GitHub

De esta forma, vemos que la expresión «nbsp» se ha traducido como «espacios duros». Esto nos da una orientación en cuanto al significado del término.

Para Ignuit, en según qué casos podían sernos útiles las referencias de los archivos «de.po» y «ru.po». De todas formas, algunas de las traducciones de estos archivos no nos parecieron del todo idóneas y no las aplicamos a nuestros *locales*.

Para la revisión de Ignuit, creamos hojas de cálculo compartidas en Google Drive a través de las cuales comentamos nuestras dudas. En total, usamos 4 archivos con los códigos: «CA», «ES», «IT» y «TODAS». A estas debemos sumar otra hoja de cálculo, la que contenía nuestro glosario propio compartido. El glosario propio fue cambiando a medida que traducíamos, pues nos fuimos dando cuenta de si unos términos determinados eran más o menos acertados o tenían un equivalente más o menos similar en el resto de lenguas de trabajo.

Las hojas de cálculo para dudas seguían el patrón que se muestra a continuación:

1	N.º segmento	Comentarios / dudas	MGP	EU	MPF
2	xxx	IGNUIT: formato --> "IGNUIT" o "Ignuit"???	En la interfaz podríamos usar "IGNUIT", mientras que en la documentación, siguiendo el original, podríamos usar "Ignuit", porque uno sería como un logo y el otro lo trataríamos como palabra (nombre propio del programa).		Ver la distinción que se hace en el archivo xml. La Mariona concuerda perfectamente. Sin embargo, en el xml it. Por tanto, habrá que mantener ambos usos.
3	xxx	gtk permite acentos y "ñ"???			
4	xxx	sticky flips: ¿Qué es? ¿Traducción?			
5	xxx	Segmentos con "gtk": ¿Traducibles? ¿A qué hace referencia "gtk"? ¿Podemos acentuar las palabras (ej.: "gtk-refés", "gtk-desfés")?			
6	xxx	Uso de artículos, preposiciones y conjunciones (ej.: "View card details" --> "Ver detalles tarjeta/ficha" o "Ver los detalles de la tarjeta/ficha").			
7	xxx	card = tarjeta			

Ilustración 89. Dudas sobre la traducción de Ignuit en el archivo compartido para todas las lenguas

28	313	Comprobar si la softkey "C" no se solapa con el segmento 310			
29					A partir del testing, se ha comprobado que estas * se destinan las tarjetas aprendidas y no aprendidas tarjeta de la Box 1 aprendida debe colocarse en la Box 2 conveniente traducir "Box" por caja. Por otra parte, en el segm redactado Box_1; en este caso se está haciendo referencia a un elemento interfaz. Sin embargo, en el segmento 274, aparece "box number one". E que es una aclaración sobre un elemento de la interfaz (no aparecerá o conveniente traducirlo por "caja núm. 1". Mi duda número uno / caja núm. uno". Es de suponer que completo con el fin de evitar problemas de compr
30	345	Box number	caja número X / número de caja		
31	356, 357	"Known" / "unknown" --> masculino???		Aprendida / no aprendida	Aprendida / no aprendida (la tarjeta/frase/palabra)
32	144 / 207	"Listen to fronts..."	Reproducir anversos		
33	POT 374	"Accepted names" / "column names"		Nombres aceptados?Nombres de columna?	
34	POT 385	DPI = Dots Per Inch. ¿Lo omitimos?			
		URI?????			

Ilustración 90. Dudas sobre la traducción de Ignuit en el archivo compartido para todas las lenguas (2)

E	F
<b>MPF</b>	<b>TG</b>
Ver la distinción que se hace en el archivo xml. La dinstinción que hace Mariona concuerda perfectamente. Sin embargo, en el xml también aparece i GNU it. Por tanto, habrá que mantener ambos usos.	
	en italiano lo puse como "capovolgere il FOGLIO ADESIVO"
	segun mi hay que dejarl asi la parte "gtk" y traducir lo que viene despues de "-"
	para el italiano, NO utilizaré los artículos cuando se trate de segmentos en los que utilice tiempos verbales de comando
Opino lo mismo. Si leemos en el Readme, Tim deja claro que el motivo por el que se escribe separado es como un juego de palabras: i GNU it; "gnu" aquí tendría la función de verbo.	
Yo diría "agenda"	pianificazioni

Ilustración 91. Dudas sobre la traducción de Ignuit en el archivo compartido para todas las lenguas (3)

Como se puede apreciar, anotamos el número de segmento que provocaba una duda, con la correspondiente consulta en la casilla de al lado. Más a la derecha, cada persona comentaba sus impresiones o posibles soluciones.

Los códigos «MGP», «EU», «MP» y «TG» corresponden a nuestros nombres.

En la columna correspondiente a los números de segmento, «xxx» implica que el comentario se aplica a todo el proyecto o a todo un documento. Generalmente, se trataba de «en\_US.po». Por su parte, «POT» significa que el comentario afecta al archivo «Ignuit.pot».

Todas aquellas preguntas que no podíamos resolver de ninguna manera sin consultar con el desarrollador las redactamos en un documento aparte, también en el Google Drive. Una vez hubimos recopilado unas cuantas preguntas, se las enviamos por correo. Estas tenían que ver, principalmente, con el contexto. Debemos recordar que los problemas técnicos con Ignuit

persistieron hasta los últimos días antes de entregar nuestro trabajo. Por ello, en contra de lo que esperábamos y deseábamos, la traducción de las cadenas se hizo sin poder utilizar el programa, es decir, sin poder ver cómo aparecía el texto (posición, tamaño, uso, forma de los menús, disposición de la información, significado de algunas palabras...).

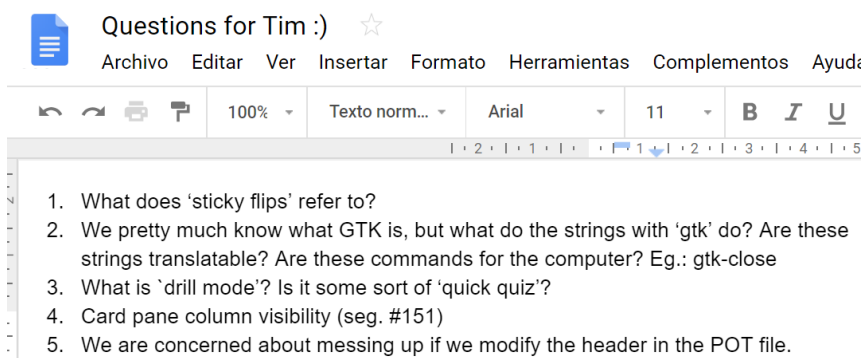


Ilustración 92. Captura del documento compartido en el que se vertieron las dudas para un posterior correo para el desarrollador

### 5.1.2. Revisiones cruzadas

Después de leer segmento por segmento la traducción propia y de ejecutar el control de calidad, enviamos la traducción a la persona correspondiente para que la revisara de nuevo con otros ojos y utilizando por segunda vez el QA de OmegaT.

Modificamos o corregimos errores ortográficos o de formato y a su vez cambiamos algunas frases para que sonaran más naturales. Por último, también corregimos errores, principalmente relacionados con el contexto, que no identificamos hasta llegados a esta fase.

### 5.1.3. Revisión conjunta

Este proceso se desempeñó solo para Iguit y solo para los archivos «en\_US.po» y «README.txt». Hicimos una corrección conjunta en la que cada uno leyó su versión por turnos para asegurarnos de que todas las traducciones fueran coherentes entre ellas.

Nos centramos principalmente en el estilo de los tres idiomas. Entre las cuestiones que discutimos se encuentra el uso de las formas verbales: ¿Imperativo o infinitivo para órdenes dirigidas a la máquina? ¿Imperativo o infinitivo cuando era la máquina la que se dirigía al usuario? ¿Qué pronombre debíamos usar? ¿Qué nivel de formalidad era más adecuado? ¿Sería en mismo en la *GUI* que en la documentación?

También detectamos diferencias en el uso de artículos y preposiciones: en lo que respecta al italiano y al español, en informática es muy común no utilizar estas partículas en determinados comandos o cadenas de texto; en catalán, en cambio, se nos hacía más difícil prescindir de estos elementos, sobre todo porque el uso del imperativo parecía demandar una frase más completa.

Usamos formas conjugadas del verbo cuando la máquina se dirigía al usuario, pero la fórmula de tratamiento no fue la misma en español que en catalán. Ej.:

Español (usted)	Catalán (vos)	Italiano (tú)
¿Desea confirmar...?	Voleu confirmar...?	Vuoi ...?

Tabla 8. Ejemplo de una cadena con un mensaje «máquina-usuario» en ES, CA e IT

Debemos decir que todavía hay aspectos a retocar en nuestras traducciones, a saber:

- Ajuste de los pronombres personales en las fórmulas de tratamiento.

- El nivel de formalidad es muy diferente en cada uno de los archivos traducidos de Iguit.

-Unificación de estrategias de traducción de referencias a otras partes del programa o a materiales de documentación.

-Unificación del uso de comillas.

En cuanto al uso de comillas, por una parte, nos costó determinar cuándo usaríamos comillas y cuándo no. Por otra parte, tuvimos que decidir qué tipo de comillas utilizaríamos. Nos dimos cuenta durante la revisión de que no habíamos sido consistentes en lo que atañe a esta cuestión.

## 5.2. Comprobación técnica

Antes de realizar el testeo, creamos los archivos traducidos y los abrimos con un editor de textos para comprobar que todo aparecía correctamente.

Para Ignuit, tal como nos había indicado el desarrollador, cambiamos los nombres de los archivos POT. Esta acción es necesaria para que la computadora sea capaz de identificar el idioma de la interfaz. Además, es imprescindible añadir los códigos de nuestras lenguas meta al archivo denominado «LINGUAS» de la carpeta de descarga. Esta tarea correspondía al desarrollador —que es quien más tarde distribuiría el *software* en las lenguas meta—. Aun así, nosotros la llevamos a cabo en nuestros ordenadores, principalmente porque nos ayudaba a asimilar el proceso de cara a nuestro futuro profesional.

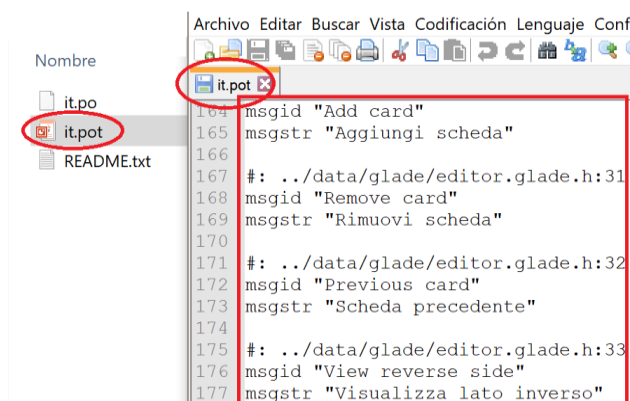


Ilustración 93. Archivo «it.pot» (traducción al IT de «ignuit.pot») en Notepad++

## 5.3. Testeo

### 5.3.1. OmegaT

El testeo de OmegaT fue una tarea relativamente sencilla. La parte positiva de este proyecto es que hemos contado con manuales en línea y con el apoyo de nuestro compañero Marc Riera.

Primero probamos de abrir OmegaT con nuestro archivo traducido desde el Símbolo de sistema de Windows (terminal), pero nos resultó imposible.

En cambio, sí funcionó en el sistema operativo alternativo de los ordenadores de la universidad: Tradumàtix (distribución de Gnu/Linux).

Las instrucciones de cómo hacerlo se encuentran en la web [How To – Translating OmegaT into your language](#).

La captura siguiente muestra las órdenes y rutas que escribimos en el terminal para ver nuestra traducción de OmegaT al catalán aplicada a la interfaz —con el programa OmegaT cerrado—:

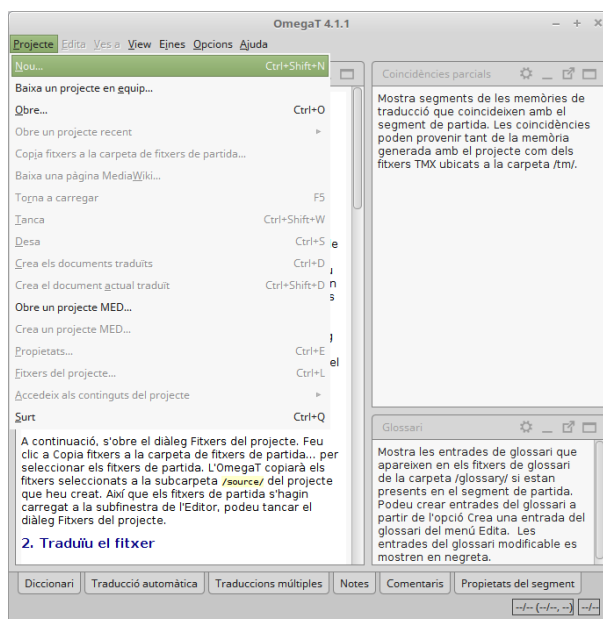
```

1333954@aulaa-01 ~/omegat/OmegaT_4.1.1
Fitxer  Edita  Visualitza  Cerca  Terminal  Ajuda
1333954@aulaa-01 ~ $ ls
Baixades  HSSstudio_UE_8_3_1_Linux_x64  Música  omegat  Públic
Documents HSTMXEditor_8_0_0_Linux_x64  odf     Open_Language_Tools  Vídeos
Escriptori Imatges                        Okapi   Plantilles
1333954@aulaa-01 ~ $ cd omegat
1333954@aulaa-01 ~/omegat $ ls
jre  OmegaT_4.1.1  OmegaT-default  plugins  scripts
1333954@aulaa-01 ~/omegat $ cd OmegaT_4.1.1
1333954@aulaa-01 ~/omegat/OmegaT_4.1.1 $ java -jar OmegaT.jar resource-bundle=/h
ome/samba/homes/1333954/Baixades/Bundle_ca.properties

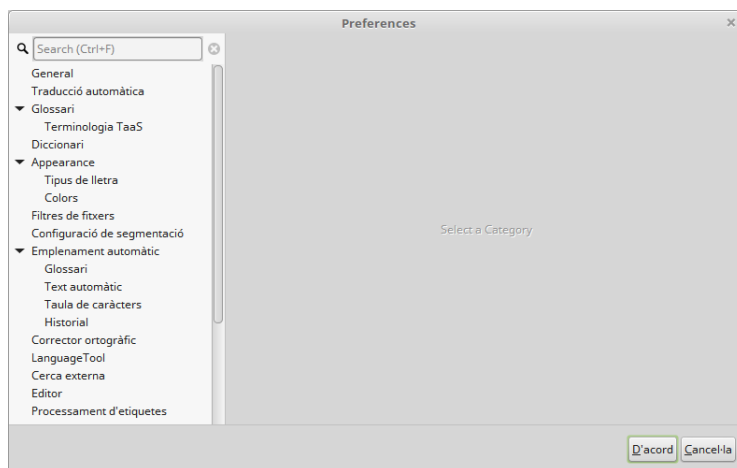
```

Il·lustració 94. Terminal del sistema operatiu Tradumàtix. Aplicació de nostres traduccions a la GUI de OmegaT

La següent vegada que obrim OmegaT, apareix la interfície amb la nostra traducció aplicada. Nos donem compte de que hi ha detalls per polir —com *hotkeys* poc apropiades, per exemple—, però, en general, el resultat era bo.



Il·lustració 95. Fase de testeo. Interfície de OmegaT traduïda al català



Il·lustració 96. Fase de testeo. Interfície de OmegaT traduïda al català (2)

Lo mismo hicimos con el español, seleccionando en la ruta del terminal el archivo correspondiente a la traducción española.

### 5.3.2. Ignuit

Para realizar el testeo de nuestra traducción, seguimos las indicaciones de Tim Musson, para lo cual fue necesario, de nuevo, recurrir a las líneas de comando del terminal en Gnu/Linux (Ubuntu):

Alternatively, you might find it more convenient to tweak the copy of ignuit you already have. That would go something like this:

- 1) Add your .po files to the ignuit-2.24.3/po directory.
- 2) Edit the file named ignuit-2.24.3/po/LINGUAS, adding the names of your new .po files.
- 3) Back in the the main source directory (i.e., ignuit-2.24.3), run the following commands:

```
$ make clean
$ ./configure --disable-scrollkeeper
$ make
$ sudo make install-strip
```

If that all goes without problems, it should now be possible to check your translations in a couple of different ways.

- 1) In your Linux desktop preferences, set your default language to the one you want to test, then launch ignuit.

- 2) To test without switching desktop language, run ignuit from a terminal window like this:

```
$ LANGUAGE=es ignuit
```

Ilustración 97. Captura de un correo del desarrollador de Ignuit con instrucciones sobre el testeo

Estos serían algunos ejemplos de cómo se ve la interfaz traducida al italiano:

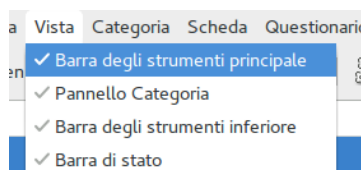


Ilustración 98. Fase de testeo. Nuestra traducción del menú «Vista» («View») en la GUI de Ignuit en IT

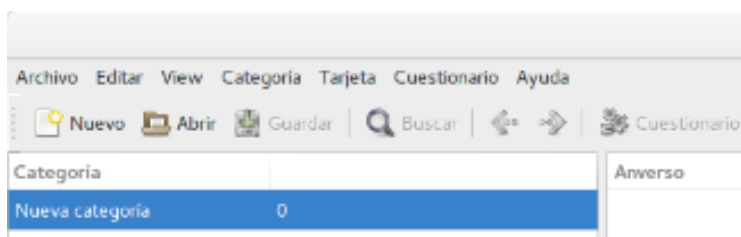


Ilustración 99. Fase de testeo. Interfaz de Ignuit traducida al español

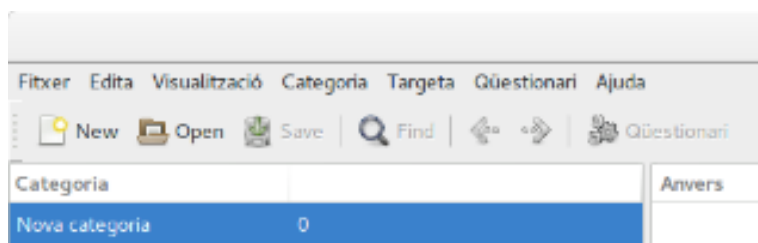
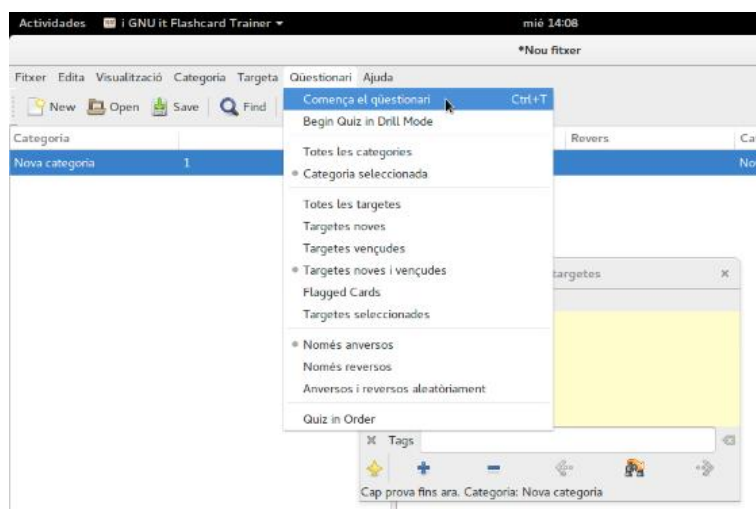


Ilustración 100. Fase de testeo. Interfaz de Ignuit traducida al catalán



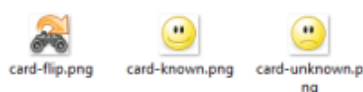
Il·lustració 101. Fase de testeo. Interfaz de Ignuit traducida al catalán (2)

Al abrirse la interfaz traducida, apreciamos que en algunas partes se mantenían las construcciones originales, como si hubiéramos olvidado traducir algunos segmentos. Si bien es cierto que algunas partes habían quedado sin traducir y éramos conscientes de ello (ej.: «*sticky flips*»), había otras cadenas que no entendíamos por qué seguían apareciendo en inglés en la interfaz traducida.

Para comprobar qué había sucedido, buscamos el texto correspondiente abriendo de nuevo el proyecto en OmegaT, pero no alcanzamos a encontrar las cadenas que contenían el texto objeto de búsqueda. A fin de disipar dudas, incluso buscamos segmentos conflictivos y los volvimos a traducir, generamos los archivos traducidos nuevamente y ejecutamos los órdenes arriba mencionadas en el terminal. Aun así, el problema persistía.

Respecto a esto, nuestro profesor nos sugirió analizar todos los archivos en la carpeta de descarga de la aplicación para intentar encontrar esas cadenas de texto en archivos otros que el PO y el POT. Nos sugirió que las palabras que no se habían traducido podrían ser imágenes en vez de cadenas de texto.

Decidimos buscar archivos de imagen en la carpeta de descarga y encontramos algunos iconos, pero ninguno contenía texto.



Il·lustració 102. Archivos de imagen de la carpeta «icons» del paquete de descarga de Ignuit

Hay otros archivos que podrían ser los que contienen el texto no traducido, aunque no llegamos a confirmar nuestra teoría antes de la presentación del trabajo. En consecuencia, dejamos esta tarea como pendiente. Contactaremos con el desarrollador para que nos confirme si el «texto invisible» pertenece a imágenes localizables y para que nos indique, si es el caso, dónde se encuentran dichas imágenes.

En otra línea, comprobamos que nuestras traducciones no sobrepasaban el espacio dispuesto para cada cadena, así como el funcionamiento de las *hotkeys* en la lengua meta, empleando la combinación Alt+[letra]:



Ilustración 103. Hotkeys en la GUI de Ignuit en IT

También verificamos si las etiquetas de formato ejecutaban su función correctamente:



Ilustración 104. Segmento «Text colours» en OmegaT

Ilustración 105. Comprobación de la funcionalidad de la etiqueta para texto en negrita en la GUI de Ignuit

Tras el testeo, concluimos que todavía hay trabajo que hacer. En cualquier caso, podemos considerar que nuestra tarea ha resultado en éxito, ya que hemos podido aplicar todas nuestras traducciones a ambas herramientas y no hemos detectado errores mayores.

## 6. Generación de los materiales finales y entrega

Una vez revisadas las traducciones, el proceso de creación de los materiales traducidos fue muy sencillo, y fue el mismo para OmegaT que para Ignuit:

Abrimos OmegaT, entramos uno por uno en los archivos que habíamos traducido y seleccionamos la opción «Crear archivo actual traducido» (Ctrl+Shift+D) del menú «Proyecto». Solo utilizamos la función «Crear archivos traducidos» para el proyecto EN>CA de Ignuit, por ser el único del que íbamos a exportar todos los archivos.

Con solo un clic, se creó la versión traducida de los archivos que nos interesaban, que se originaron automáticamente en la carpeta «target» del repositorio de nuestro ordenador —no en GitHub—.

La mayoría se generaron manteniendo el mismo nombre que en el original, a excepción de algunos en los que automáticamente se añadió el [código ISO](#) de la lengua meta (ej.: «Bundle\_ca.properties»).

Para nuestra entrega, no teníamos consigna expresa de cambiar los nombres de los archivos; sin embargo, era necesario cambiar «en\_US.po» para evitar confusión y, por otra parte, para el testeo de Ignuit —según nos había explicado el desarrollador— era necesario modificar el nombre de «ignuit.pot» a «es.pot», «it.pot» y «ca.pot».

Además, como tarea complementaria a la generación de los archivos traducidos, añadimos nuestros nombres al *copyright* de OmegaT, y también nuestras lenguas meta al archivo «LINGUAS» del proyecto Ignuit. Paralelamente, modificamos la cabecera de los archivos «.po» en las lenguas meta de Ignuit a fin de incluir determinados metadatos (ej.: «Last translator»).

Para obtener la memoria resultante de nuestras traducciones, tras haber exportado todos los archivos traducidos que nos interesaban, abrimos nuestra carpeta local, en la cual se habían generado las memorias de traducción «xxx-level1.tmx», «xxx-level2.tmx» y «xxx-



omegat.tmx». Según se explica en la documentación de OmegaT<sup>15</sup>, la memoria de nivel 1 contiene solo información textual —puede abrirse en múltiples herramientas de TAO—; la de nivel 2 contiene etiquetas específicas de OmegaT que podrían ser compatibles con otras herramientas de TAO; la memoria «xxx-omegat.tmx» es exclusiva para su reutilización en proyectos de OmegaT. Las memorias que entregamos a nuestro profesor fueron las del tipo «omegat», puesto que él trabaja con esta aplicación y, por tanto, podía abrir perfectamente los archivos.

Ahora ya teníamos nuestros archivos en las lenguas meta y nuestras memorias de traducción resultantes. Sin embargo, nos faltaba un paso: filtrar, en las memorias del proyecto OmegaT, solo los segmentos que habíamos modificado nosotros, sin incluir traducciones de otros usuarios.

Según la información que teníamos, esto era posible a través de un *script* incorporado en OmegaT. Los *scripts* son funciones que se activan a través de un programa pero «llaman» a otro programa para que las ejecute. Pensábamos pedir ayuda a Kos Ivantsov, pero antes de llegar a ese paso, Marc Riera nos explicó que podíamos evitarnos el paso del *script* y utilizar en su lugar la herramienta libre Olifant, desarrollada por [Okapi Framework](http://Okapi Framework).

No teníamos Okapi Olifant instalado en ninguno de nuestros ordenadores, aunque eso no fue problema, ya que se puede descargar rápidamente desde la red. Para instalarlo, simplemente debemos seguir los pasos del asistente de instalación.



Ilustración 106. Descarga de Olifant desde Source Forge

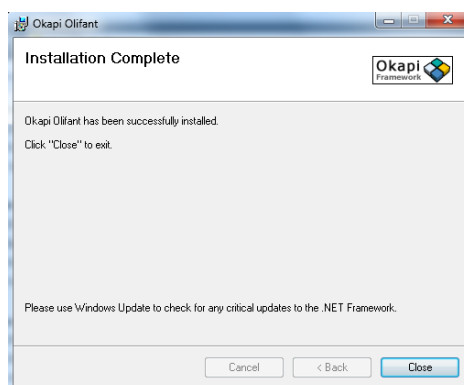


Ilustración 107. Asistente de instalación de Olifant. Instalación completada

<sup>15</sup> Véase el capítulo 13 del manual de ayuda de OmegaT en línea: <https://omegat.sourceforge.io/manual-latest/en/chapter.translation.memories.html>

- Abrir nuestra memoria TMX desde Okapi Olifant.

- Abrir nuestra memoria TMX desde Okapi Olifant.

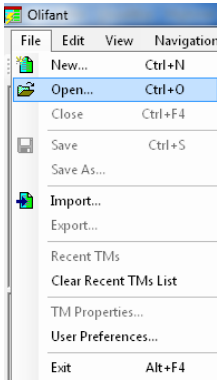


Ilustración 108. Menú «File» en Olifant para abrir nuestra MT

Se mostrará de la siguiente manera:

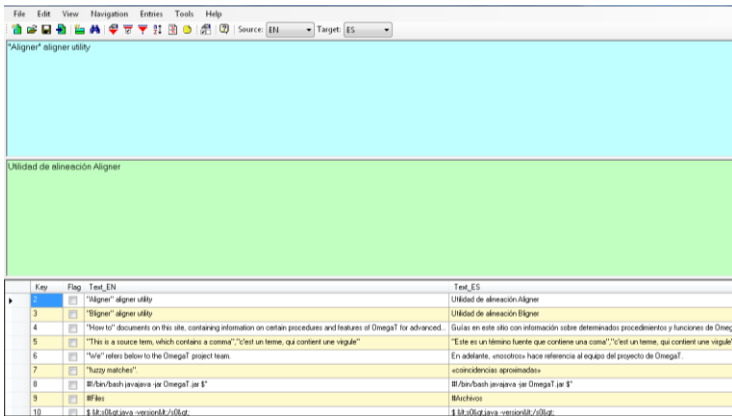


Ilustración 109. Interfaz de Olifant con nuestra MT del proyecto EN>ES de OmegaT abierta

- En el menú «*View*», seleccionar «*Fields Settings*» para determinar los campos que queremos que se muestren en pantalla.

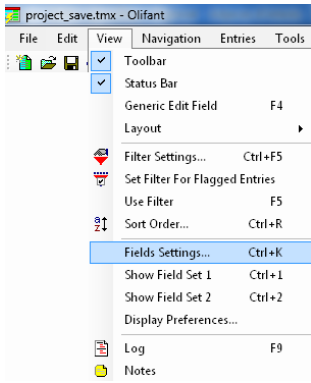


Ilustración 110. Menú «View» en Olifant. Selección de la función «Fields Settings»

En nuestro caso, seleccionamos «ChgUser\_XX» y «ChgDate\_XX», donde «XX» corresponde a la lengua meta. Estos campos hacen referencia al último usuario que modificó un determinado segmento en lengua meta (ES en la captura) y a la fecha de dicha modificación, respectivamente.

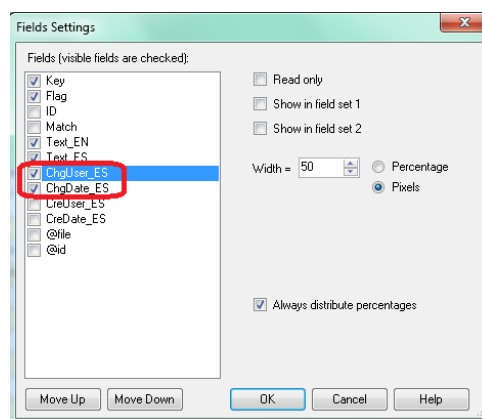


Ilustración 111. Elección de los metadatos de la MT que queremos que muestre Olifant

- Al pulsar «OK», en la ventana principal veremos el cambio:

	Key	Flag	Text_EN	Text_ES	ChgUser	ChgDate
▶	2	<input type="checkbox"/>	"Aligner" aligner utility	Utilidad de alineación Aligner	Cristina	16/09...
	3	<input type="checkbox"/>	"Bligner" aligner utility	Utilidad de alineación Bligner	Cristina	16/09...
	4	<input type="checkbox"/>	"How to" documents on this site, containing information ...	Guías en este sitio con información sobre determinados proce...	Cristina	17/09...
	5	<input type="checkbox"/>	"This is a source term, which contains a comma"."c'est ...	"Este es un término fuente que contiene una coma"."c'est un ...	taty	16/11...
	6	<input type="checkbox"/>	"We" refers below to the OmegaT project team.	En adelante, «nosotros» hace referencia al equipo del proyect...	Cristina	03/10...
	7	<input type="checkbox"/>	"fuzzy matches".	«coincidencias aproximadas»	hcarta...	01/12...
	8	<input type="checkbox"/>	#!/bin/bash javajava -jar OmegaT.jar \$"	#!/bin/bash javajava -jar OmegaT.jar \$"	Antonio	30/03...
	9	<input type="checkbox"/>	#Files	#Archivos	hcarta...	17/11...
	10	<input type="checkbox"/>	\$ &lt;s0&gt;java -version&lt;/s0&gt;	\$ &lt;s0&gt;java -version&lt;/s0&gt;	Cristina	22/08...
	11	<input type="checkbox"/>	\$(adjustedScore)	\$(adjustedScore)	eumeu	16/11...
	12	<input type="checkbox"/>	\$(changedDate)	\$(changedDate)	taty	05/12...

Ilustración 112. Vista de la MT del proyecto EN>ES de OmegaT con los metadatos seleccionados

- En la barra de herramientas, hacer clic en el botón correspondiente a «Filter Settings».

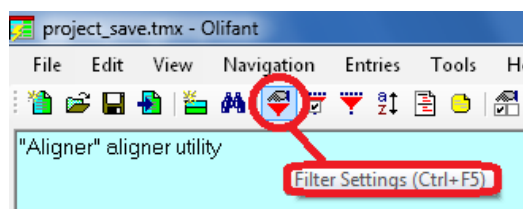


Ilustración 113. Botón «Filter Settings» de Olifant

Esta función permite filtrar segmentos determinados de la MT.

Podemos filtrar segmentos marcados («flagged») o segmentos que cumplan una determinada condición. Si seleccionamos la segunda opción, para poder indicar al programa cuál es la condición, necesitamos conocer la sintaxis de la orden. No sabíamos cuál era esa sintaxis, pero vimos en el menú «Ejemplos» («Examples») que aparecía una lista con posibles casos, y uno de ellos era precisamente el que nos interesaba. En la imagen que sigue se ve la lista de ejemplos propuestos. El último elemento coincide con la condición que queríamos definir: «Target entry was last changed by 'XYZ'». Esta propiedad indica que se mostrarán solo los segmentos cuya última modificación haya sido realizada por un usuario con una ID específica. Esta ID es la variable que se escribiría en el lugar de «XYZ».

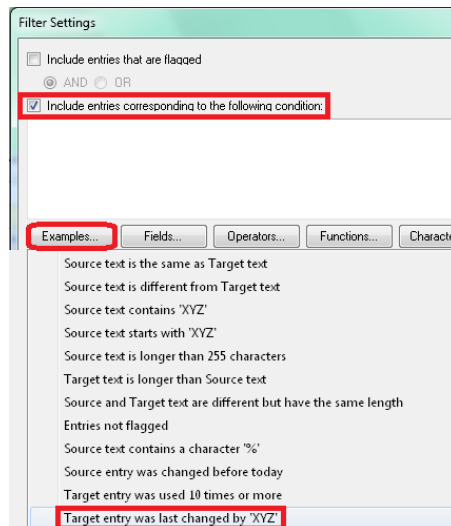


Ilustración 114. Interfaz de Olifant. Filtro para seleccionar autoraje del último cambio en la MT

- Al seleccionar la condición «*Target entry was last changed by 'XYZ'*», aparece en el cuadro un comando: «*ChgUser\_XX = 'XYZ'*». Esta era, pues, la sintaxis para la orden que queríamos dar al programa. Solo teníamos que sustituir «XYZ» por nuestras ID de usuario. Como para cada lengua queríamos indicar más de una ID, necesitábamos encontrar también un nexo para la sintaxis de nuestra orden entre nombres de usuarios. Haciendo pruebas, acabamos dando con la fórmula que nos hacía falta: «*ChgUser\_XX = 'XYZ' OR ChgUser\_XX = 'XYZ'*».

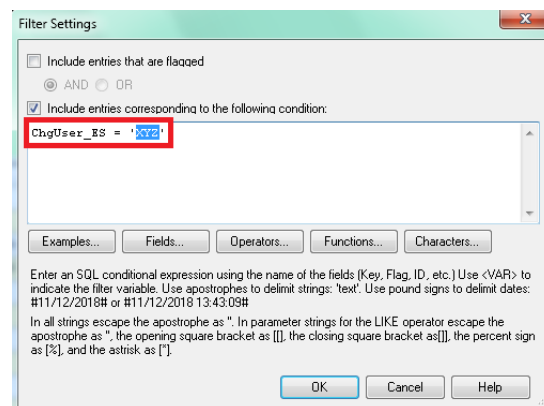


Ilustración 115. Fórmula «ChgUser\_ES = 'XYZ'» en Olifant

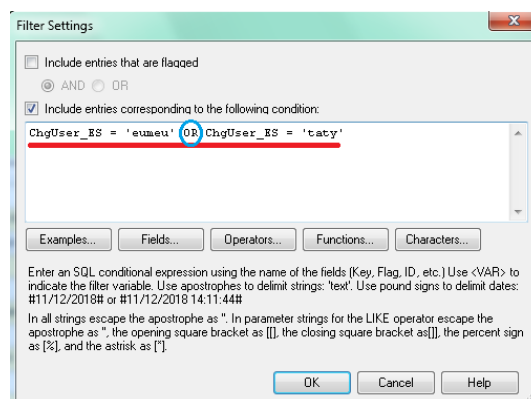


Ilustración 116. Selección de múltiples ID de usuario para la función «ChgUser\_ES = 'XYZ'» en Olifant

- Llegados a este punto, solo se muestran los segmentos de la MT cuya última modificación fue realizada por alguno de nosotros.

Key	Flag	Text_EN	Text_ES	ChgUser	ChgDate
4316	<input type="checkbox"/>	maximizes the pane	maximiza el panel	eumeu	26/11...
4370	<input type="checkbox"/>	number of segments - translated vs total for the current file	número de segmentos - traducidos vs el total del archivo actual	eumeu	16/11...
4372	<input type="checkbox"/>	number of unique segments - translated vs total in the pr...	número de segmentos únicos - traducidos vs segmentos totale...	eumeu	30/11...
4452	<input type="checkbox"/>	restores the layout before the maximizing step	restaura la disposición antes de maximizar la ventana	eumeu	26/11...
4532	<input type="checkbox"/>	the list of all translatable files in the project.	la lista de todos los archivos traducibles en el proyecto.	eumeu	16/11...
4552	<input type="checkbox"/>	total number of segments (including repeats) in the project	número total de segmentos en el proyecto (repeticiones incluid...	eumeu	16/11...
4579	<input type="checkbox"/>	undocks the pane from the main window	desbloquea el panel desde la ventana principal	eumeu	26/11...
5	<input type="checkbox"/>	"This is a source term, which contains a comma","c'est ...	"Este es un término fuente que contiene una coma","c'est un ...	taty	16/11...
12	<input type="checkbox"/>	\${changedDate}	\${changedDate}	taty	05/12...
14	<input type="checkbox"/>	\${diffReversed}	\${diffReversed}	taty	05/12...
16	<input type="checkbox"/>	\${fileNameOnly}	\${fileNameOnly}	taty	05/12...
23	<input type="checkbox"/>	\${noStemScore}	\${noStemScore}	taty	05/12...
24	<input type="checkbox"/>	\${score}	\${score}	taty	05/12...

Ilustración 117. MT del proyecto EN>ES de OmegaT únicamente con segmentos modificados por las ID de usuario «eumeu» y «taty» (Eugenia y Tatiana)

- El último paso es exportar esta MT filtrada desde «File > Export».
- Aparece una ventana en la que debemos seleccionar una serie de datos. Marcamos los que nos convengan y aceptamos.

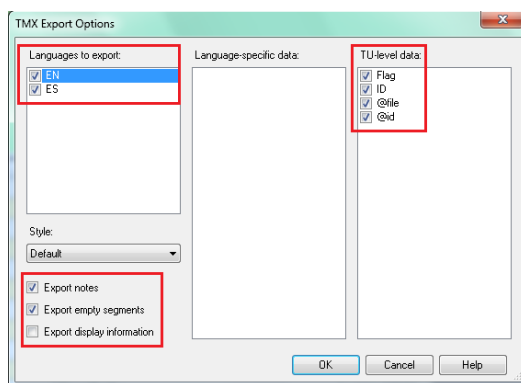


Ilustración 118. Opciones de exportación de la MT desde el asistente de Olifant

- Por último, solo hace falta guardar la MT filtrada en la carpeta que nos interese con un nombre de archivo.

## 7. Conclusiones

En primer lugar, esta era la primera vez que hacíamos una traducción de este tipo. El tener que gestionar nuestro propio trabajo fue un reto: aunque hemos recibido ayuda externa, no teníamos a ninguna persona u organización que nos marcara unas pautas o nos indicara cómo proceder, lo cual nos dio mucha libertad, pero a la vez representó una dificultad porque nunca habíamos hecho nada parecido y teníamos mucho que aprender. Nos dimos cuenta, no obstante, de que el trabajo en equipo tiene un enorme potencial; si se gestiona bien se pueden lograr grandes cosas. Hubo buena comunicación y apoyo mutuo entre nosotros, lo que fue esencial para el desarrollo del proyecto.

Del mismo modo, la comunicación con los desarrolladores fue indudablemente imprescindible. El proceso habría sido infinitamente más complicado —o quizás incluso imposible— sin su ayuda. Tuvimos la suerte de que tanto el creador de OmegaT como los responsables de la localización de OmegaT estuvieran disponibles y siempre atentos al progreso del trabajo.

En relación con la profesión de traductores, una pregunta que nos surgió es la siguiente: ¿Cómo se organizan las grandes empresas para conseguir que haya consistencia entre traducciones de un mismo producto a varias lenguas? Para ejemplificar la pregunta: si los desarrolladores de un producto quieren lanzar versiones para varios *locales* diferentes, ¿cómo se logra la coherencia entre las múltiples adaptaciones a nivel lingüístico? Entendemos

que hay muchas diferencias insalvables, pero sabemos que es conveniente que, en un programa como puede ser Iguit, si usamos el término «tarjeta» en español, usemos «targeta» —y no «fitxa», por ejemplo— en catalán. Esta fue una de nuestras mayores dificultades: el consenso y la consistencia. Aun a fecha de hoy, hay ciertos ajustes que debemos aplicar a nuestras traducciones para que se acoplen mejor unas con las otras, sobre todo en Iguit —en OmegaT podemos consultar versiones anteriores del *software*, que nos dan mucha información útil en este sentido—. Nuestra idea es seguir trabajando en este proyecto aun cuando termine el periodo lectivo de TPD para alinear nuestras propias traducciones con el mismo OmegaT a fin de realizar mejoras relacionadas con la consistencia entre lenguas meta. Otro paso primordial a completar es el de «pulir» el glosario propio hasta llegar a tener una versión homogénea que podamos dar por definitiva.

Nos hemos dado cuenta a través de nuestras investigaciones de que la localización es una profesión emergente; en los últimos años, se ha creado un sinfín de oportunidades. A pesar de ello, no es un trabajo fácil: requiere de una serie de conocimientos complementarios a la traducción en sí. Gracias a esta tarea, hemos podido adquirir más práctica y confianza. La formación dentro del campo de la localización (gestión de formatos, ingeniería, etc.) puede representar para los traductores un impulso a nivel laboral. Para nosotros, de todas formas, lo más relevante fue la sensación de satisfacción que sentimos al ver nuestras traducciones aplicadas a los programas en la fase de testeo. Los avances que ha vivido nuestra generación en relación con la informática nos han llevado a poder experimentar este suceso que, unos años antes, nos habría parecido producido por arte de magia.

Una de las cuestiones que quedan en el tintero es si existe alguna manera de evitar los problemas de falsos positivos generados con el control de calidad de OmegaT. OmegaT no es capaz de identificar lematizaciones de términos; por tanto, no puede detectar variaciones morfológicas de una misma palabra y las muestra como errores de traducción. De ahí que al ejecutar el control de errores aparecieran más de 300 que, en realidad, eran falsos positivos. Nos ha faltado investigar si existe algún *script* que permita lidiar con este problema, o, en caso de que no exista ninguno aún, si se podría crear. Como última instancia, podríamos proponer al equipo de desarrollo del programa que se planteara mejoras del *software* para evitar a los localizadores una molestia innecesaria. Desconocemos si el uso del [servicio de terminología TaaS](#), con el que OmegaT se puede conectar, habría resuelto el problema. Nuestro límite de tiempo para completar el trabajo nos impidió investigar más sobre esta cuestión.

Reunimos aquí algunas de las lecciones que hemos aprendido y las aptitudes que hemos adquirido:

- Reciclando lo estudiado en las asignaturas del Máster, hemos tenido que profundizar a la hora de llevar a cabo las tareas que nos correspondían. Aplicamos conocimientos de muchas de ellas, si no todas: Trados, MemoQ, DéjàVu X, OmegaT, Introducción a la localización, Fundamentos de la tradumática, Bases de datos terminológicas, Ingeniería de la localización, Gestión de formatos, SGET, Sistemas operativos libres, Fundamentos de la localización y la TA, Catalyst, Corpus, Memsource, las sesiones de TPD... Nos habría gustado cursar antes Expresiones regulares, ya que hemos necesitado hacer uso de esta técnica en ocasiones y no hemos podido sacar provecho de lo que intuimos que se imparte en esta asignatura.

- La colaboración nos ha permitido aprender todos de todos, así como de lo que tantas otras personas pueden enseñarnos.

- Como es evidente, hemos aprendido a utilizar OmegaT, incluyendo muchas de sus funciones y atajos de teclado.

- Ahora sabemos mucho más sobre *software* libre, ingeniería y gestión de proyectos de traducción en equipo, y esperamos seguir avanzando en nuestra formación.

- Las competencias en localización se han hecho vitales para destacar como traductores en el mundo actual.

-La localización de *software* tiene muchas particularidades. Como profesionales, no solamente necesitamos una sensibilidad especial hacia las lenguas y su uso, sino que a la vez debemos dominar todas las características que rodean esta tipología de texto en concreto.

En el momento de entregar nuestro trabajo, podemos decir que:

-La interfaz de OmegaT ha quedado traducida hasta el 70,9% en CA —contando las cadenas que ya se habían traducido previamente—, mientras que hemos contribuido también a la traducción de cadenas de texto de la documentación, tanto en catalán como en español. Nuestra aportación ha llegado a los desarrolladores gracias a GitHub, y esperamos que las traducciones sean aceptadas y aplicadas a posteriores versiones del programa.

-La interfaz de Ignuit está traducida prácticamente en su totalidad en IT, ES y CA, así como también hemos colaborado con cadenas de texto de otros archivos. Nuestras traducciones no se han hecho públicas porque nos gustaría resolver determinadas dudas y acabar de perfeccionar nuestro trabajo. A pesar de esto, Tim Musson accedió a añadir nuestros *locales* a la carpeta de descarga de Ignuit una vez la interfaz esté terminada.

Algunos temas han quedado pendientes en el momento de presentar esta memoria escrita. Además, la traducción completa está algo lejos de llegar a su fin para ambos programas; sin embargo, estamos orgullosos de haber podido aportar nuestra parte y, como hemos dicho anteriormente, tenemos intención de dar continuidad a este proyecto más allá de la asignatura TPD.

Nos habría gustado usar en el proceso más herramientas de *software* libre. Usamos las que mejor conocíamos —a excepción de alguna que descubrimos a raíz del trabajo—; las que conocíamos con anterioridad eran, en su mayoría de licencia privativa.

Respecto a nuestra decisión de localizar dos programas, la consideramos positiva porque el potencial de aprendizaje se hizo así más grande; a la vez, ha jugado un poco en nuestra contra porque ha supuesto un volumen de trabajo mayor al que habíamos calculado. Además, no hemos podido concentrar todos nuestros esfuerzos en una sola cosa.

Seguidamente, exponemos una lista de las ventajas y desventajas más relevantes de traducir con OmegaT según nuestra experiencia:

Ventajas	Desventajas
<i>Software</i> libre y gratuito.	La interfaz tiene muchas pestañas; mostrarlas todas no es muy viable y acoplarlas a la ventana del Editor puede hacernos pasar por alto detalles importantes.
Multiplataforma (se adapta a cualquier sistema operativo) con el requisito de la instalación en Windows de la plataforma JAVA.	No es eficaz para la constitución de glosarios.
Fácil de instalar y de usar. La estructura de carpetas de los proyectos, que se crea automáticamente, facilita la gestión de recursos (memorias de traducción de referencia, automáticas o penalizadas, glosarios, archivos de partida y en lengua meta, etc.). No hay que crear ningún paquete de proyecto.	Por el diseño de la interfaz y según la configuración que se use, algunos segmentos sin traducir pueden pasar inadvertidos.
Se puede conectar con repositorios en GitHub que hicieron cómodo el trabajo en equipo.	No permite crear proyectos multilingües.
Usa formatos estándar.	Genera falsos positivos al no detectar la

	lematización de palabras.
Conserva el formato de los archivos, a diferencia de otras herramientas, como Trados.	No tiene configurados los nuevos diacríticos del catalán (se puede resolver configurando la herramienta de corrección ortográfica)
Tiene una comunidad activa de desarrolladores y usuarios a quienes consultar en caso de dudas o problemas.	Las etiquetas de formato no se gestionan de la manera más práctica. Así como otras aplicaciones muestran directamente el texto con el formato aplicado, en OmegaT figura el texto normal con etiquetas como “</b>”.
Ofrece la posibilidad de diferenciar mediante distintos colores los segmentos traducidos de los que no lo están, entre otras opciones.	La función de autopropagación no funcionó para algunos archivos del proyecto Ignuit.
Permite configurar la búsqueda externa, la función de autocompletar texto y el texto predictivo.	Como SGET, no ofrece tantas posibilidades como otras herramientas.
Se puede conectar a motores de TA (algunos requieren una clave API).	No tiene función de extracción terminológica (pero se puede conectar al servicio de terminología TaaS, el cual no usamos).
Muchas utilidades se pueden ejecutar con atajos de teclado.	En el Editor no se puede previsualizar el resultado de la traducción.
Soporta el UTF-8.	No permite crear los archivos traducidos en formato bilingüe.
Incluye filtros para multitud de tipos de archivo, incluyendo «.po» y «.pot».	Algunos formatos de archivo no son soportados, aunque este inconveniente no afectó a nuestro caso.
Permite alinear textos.	
Permite incorporar más de una memoria de traducción a un mismo proyecto.	
Crear y compartir proyectos es muy sencillo.	

Tabla 9. Ventajas e inconvenientes de traducir con OmegaT

Por último, solo nos queda decir que esta ha sido una fantástica oportunidad de aprendizaje. Esta experiencia nos ha abierto las puertas de un mundo muy interesante. Los cuatro estamos de acuerdo en que este primer contacto con la traducción de productos digitales no ha hecho sino motivarnos a seguir aprendiendo y a querer descubrir más sobre la disciplina de la localización. Nuestra curiosidad ha crecido con este desafío que consideramos apasionante y muy enriquecedor.



## 8. Bibliografía

- Agar, M. (2006). International Journal of Qualitative Methods, 5(2), (2006). *Culture: Can you take it anywhere?* [En línea] [http://www.ualberta.ca/~iiqm/backissues/5\\_2/pdf/agar.pdf](http://www.ualberta.ca/~iiqm/backissues/5_2/pdf/agar.pdf) [Consultado el 05/12/2018]
- Anayet formación. (2014). *El sistema Leitner. La memoria, 3ª parte*. [En línea] <https://www.academiazaragoza.com/single-post/2014/03/30/El-Sistema-Leitner-La-memoria-3%C2%AA-parte> [Consultado el 29/10/2018]
- Coromines, Joan, (1989). *Lleures i converses d'un filòleg*. Barcelona: Club Editor.
- OmegaT Documentation. *User manual* [En línea] <http://omegat.org/documentation> [Consultado el 02/10/2018]
- Doğru, G., Martín-Mor, A., Ortiz-Rojas, S. (2017). *MTradumàtica: Free Statistical Machine Translation Customisation for Translators* [En línea] [https://ufal.mff.cuni.cz/eamt2017/user-project-product-papers/papers/user/EAMT2017\\_paper\\_75.pdf](https://ufal.mff.cuni.cz/eamt2017/user-project-product-papers/papers/user/EAMT2017_paper_75.pdf)
- D'Itri M., Potortì F. (2005). Linux. Free Translation Project - Sezione Italiana. *Glossario di termini tecnici per traduttori di programmi liberi* [En línea] <http://tp.linux.it/glossario.html> [Consultado el 01/11/2018]
- El sistema operativo GNU. (2018). *¿Qué es el software libre?* [En línea] <https://www.gnu.org/philosophy/free-sw.es.html> [Consultado el 05/12/2018]
- ÉsAdir (2018). *Accent diacrític*. [En línea] <http://esadir.cat/gramatica/criteris/diacritics> [Consultado el 06/11/2018]
- F. Serrador, F. J., Castro, X., Corral, L. et al. (2006). *Localización de GNOME al español*. Openshine S.L. [En línea] <https://mail.gnome.org/archives/gnome-es-list/2011-October/pdfwqsWlw3tLC.pdf> [Consultado el 29/10/2018]
- Fabra, Pompeu (1956). *Gramàtica*. Ed. Teide.
- FileInfo. (2018). *The File Extensions Database*. [En línea] <https://fileinfo.com/> [Consultado el 30/10/2018]
- Free Software Foundation, Inc. (2008). *GNU Free Documentation License*. [En línea] <https://www.gnu.org/licenses/fdl-1.3.en.html> [Consultado el 28/10/2018]
- GALA Globalization & Localization Association. (2018). *What is internationalization?* [En línea] <https://www.gala-global.org/language-industry/intro-language-industry/what-internationalization> [Consultado el 10/11/2018]
- Gil Castro V. E., Rosas Armenta X., Olmeda Quiroz M., (2013). *Fundamentos de las tecnologías de información: viviendo en una sociedad tecnológica*, Volumen I. México: Editorial Digital Tecnológico de Monterrey. [En línea] <https://books.google.es/books?id=vUJuDQAAQBAJ&printsec=frontcover&dq=tecnologias+en+el+mundo+actual&hl=ca&sa=X&ved=0ahUKewjSstCCwI3fAhXJIsAKHYtxAkQQ6AEIXTAH#v=onepage&q=tecnologias%20en%20el%20mundo%20actual&f=false> [Consultado el 05/12/2018]
- GNOME Damned Lies. (2016-2018). [En línea] <https://l10n.gnome.org/> [Consultado el 29/10/2018]
- GNOME. (2018). *Gtranslator* [En línea] <http://projects.gnome.org/gtranslator> [Consultado el 29/10/2018]
- GNU (2018). *GNU gettext utilities*. [En línea] <https://www.gnu.org/software/gettext/manual/gettext.html> [Consultado el 29/10/2018]

- GNU Operating System. (2007). *GNU General Public License*.  
<https://www.gnu.org/licenses/gpl.html>
- GNU Operating System (2016). *Gettext* [en línea]  
<https://www.gnu.org/software/gettext/> [Consultado el 20/01/2019]
- GUL UC3M. *Nos gusta jugar con la tecnología*. [En línea] <http://ww.gul.es> [Consultado el 29/10/2018]
- How to. *Translating OmegaT into your own language*. [En línea]  
[http://omegat.org/howtos/localizing\\_omegat](http://omegat.org/howtos/localizing_omegat) [Consultado el 02/10/2018]
- KDE UserBase Wiki. *Lokalize*. [En línea] <https://userbase.kde.org/Lokalize> [Consultado el 29/10/2018]
- L. Molina y A. Hurtado Albir. (2002) *Translation Techniques Revisited: A Dynamic a Functionalist Approach*, Meta 47(4), 498- 512. [En línea]  
<http://www.erudit.org/revue/meta/2002/v47/n4/008033ar.html?vue=resume> [Consultado el 05/12/2018]
- La Creu, J.; Martínez, N. (1993). *Vocabulari d'informàtica*. Generalitat Valenciana, Conselleria d'Educació i Ciència. Direcció General de Política Lingüística: Colomar Editors. [En línea]  
<http://www.ceice.gva.es/documents/161863176/162975848/Inform%C3%A1tica.pdf/8d20b908-e4ca-4c6f-b33d-e614ec589e13> [Consultado el 01/11/2018]
- Luca.diligander.libero.it. *Linee guida per la traduzione di GNOME*. [En línea]  
<https://digilander.libero.it/elleuca/files/linee-guida.pdf> [Consultado el 29/10/2018]
- Martín-Mor, A., Piqué, R. (2017). *MTradumàtica i la formació de traductors en Traducció Automàtica Estadística* [En línea]  
[https://ddd.uab.cat/pub/tradumatica/tradumatica\\_a2017n15/tradumatica\\_a2017n15p97.pdf](https://ddd.uab.cat/pub/tradumatica/tradumatica_a2017n15/tradumatica_a2017n15p97.pdf) [Consultado el 28/10/2018]
- Martin-Mor, A.; Piqué Huerta R.; Sánchez Gijón, P. (2016). *Tradumàtica: tecnologies de la traducció*.
- Memoq Help. *Pre-Translation*. [En línea]  
<http://memoq.helpmax.net/en/explanations/productivity-features/pre-translation/> [Consultado el 01/11/2018]
- Moreno Sorona, L. (2018). *A comparative study of OmegaT and Memsource: advantages and disadvantages of their primary features*. UOC Universitat Oberta de Catalunya [En línea]  
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/74665/3/lmorenoTFM0118memoria.pdf> [Consultado el 03/11/2018]
- Notepad++ (2018) [En línea] <https://notepad-plus-plus.org/>
- Optimot. *Consultes lingüístiques*. [En línea]  
<http://aplicacions.llengua.gencat.cat/llc/AppJava/index.html> [Consultado el 01/11/2018]
- Piqué, R.; Sánchez-Gijón. (2006). *La digitalización del proceso de traducción: tareas y herramientas*. Dipòsit general de documents de la UAB [En línea]  
<https://ddd.uab.cat/pub/poncom/2006/169768/tareas-herramientas-argentina.pdf> [Consultado el 04/11/2018]
- Quizlet. (2018) *Studying with Flashcards mode*. [En línea]  
<https://quizlet.com/it/help/2444092/study-with-flashcards> [Consultado el 02/11/2018]

Raymond.cc computers, made easy. (2017). *5 Tools To Help Identify Unrecognized or Unknown File Types*. [En línea] <https://www.raymond.cc/blog/what-file-is-that-how-to-identify-unrecognized-files-types-and-format/> [Consultado el 30/10/2018]

Sens dubte. (2018). Gestor de consultes lingüístiques i terminològiques. *Non-breaking space, hard space*. [En línea] [http://www2.ub.edu/assessling/cgi/consultes/resposta.pl?consultes\\_id=1064](http://www2.ub.edu/assessling/cgi/consultes/resposta.pl?consultes_id=1064) [Consultado el 01/11/2018]

Sharelatex (2018) [En línea] <https://www.sharelatex.com/>

Sherman, L. (1998). TechWhirl Archives. *Pane or Panel?* [En línea] [http://www.techwhirl.com/archives/9811/techwhirl-9811-00780.html#.W\\_w2bzGCGUk](http://www.techwhirl.com/archives/9811/techwhirl-9811-00780.html#.W_w2bzGCGUk) [Consultado el 21/11/2018]

Signs and Symptoms of Translation. (2018). *Penalties and filters in translation memories* [En línea] <https://signsandsymptomsoftranslation.com/2018/04/03/penalties/> [Consultado el 01/11/2018]

Stallman, Richard M. (1996-2018). *Richard Stallman's Personal Site* [En línea] <https://stallman.org/> [Consultado el 05/12/2018]

Stallman, Richard M. (2004). *Software libre para una sociedad libre*. [En línea] [https://www.gnu.org/philosophy/fsfs/free\\_software.es.pdf](https://www.gnu.org/philosophy/fsfs/free_software.es.pdf) [Consultado el 05/12/2018]

Stallman, R. (2013). *¿Código abierto es lo mismo que software libre?* [En línea] <http://www.cobdc.net/programarilliure/codigo-abierto-es-lo-mismo-que-software-libre/> [Consultado el 05/12/2018]

SoftCatalà. (2000-2001). *Guia d'estil per a la localització de programari*. [En línea] <http://punttic.gencat.cat/sites/default/files/MaterialsFormatius/32/html/guia/guia.html> [Consultado el 29/10/2018]

Softcatalà. *Glossari*. En: Projectes/GNOME [En línea] <https://www.softcatala.org/wiki/Projectes/GNOME/Glossari> [Consultado el 29/10/2018]

Sourceforge. (2018). [En línea] <https://sourceforge.net>

TERMCAT. (2018). *Centre de terminologia*. [En línea] <http://www.termcat.cat/> [Consultado el 01/11/2018]

The Document Foundation Wiki (2018). *Guía de traducción y estilo Libre Office en español*. [En línea] [https://wiki.documentfoundation.org/ES/Gu%C3%ADa\\_de\\_estilo](https://wiki.documentfoundation.org/ES/Gu%C3%ADa_de_estilo) [Consultado el 29/10/2018]

Traductores Oficiales (2015). *La localización: traducción de software y páginas web* [En línea] <http://traductoresoficiales.es/es/la-localizacion-traduccion-de-software-y-paginas-web/>

Universia Argentina. (2018). *Lenguajes de programación populares*. [En línea] <http://noticias.universia.com.ar/consejos-profesionales/noticia/2016/02/22/1136443/conoce-que-lenguajes-programacion-populares.html> [Consultado el 05/12/2018]

Università Mediterranea (2018). *La patente europea del computer. Glossario dei termini informatici più comuni* [En línea] <http://www.artec.unirc.it/pac/glossario.pdf> [Consultado el 06/11/2018]

Ubuntu wiki (2018). *Estilo*. [En línea] <https://wiki.ubuntu.com/UbuntuSpanishTranslators/Estilo> [Consultado el 29/10/2018]

Varios autores (2019). *Speaking with OmegaT's Project Manager*. THE ATA CHRONICLE [En línea] <http://www.atanet.org/chronicle-online/highlights/speaking-with-omegats-project-manager/#sthash.2HZZzmkkm.dpbs> [Consultado el 02/11/2018]

Wikipedia. (2018). *Código fuente*. [En línea] [https://es.wikipedia.org/wiki/C%C3%B3digo\\_fuente](https://es.wikipedia.org/wiki/C%C3%B3digo_fuente) [Consultado el 05/12/2018]

Wikipedia. (2018). *Flashcards*. [En línea] [https://es.wikipedia.org/wiki/Flash\\_cards](https://es.wikipedia.org/wiki/Flash_cards) [Consultado el 29/10/2018]

Wikipedia. (2018). *GNU*. [En línea] <https://ca.wikipedia.org/wiki/GNU> [Consultado el 29/10/2018]

Wikipedia (2018). *Inserció d'espais protegits, de guionets i de separadors condicionals*. [En línea] [https://help.libreoffice.org/Common/Inserting\\_Protected\\_Spaces,\\_Hyphens\\_and\\_Conditional\\_Separators/ca](https://help.libreoffice.org/Common/Inserting_Protected_Spaces,_Hyphens_and_Conditional_Separators/ca) [Consultado el 01/11/2018]

Wikipedia. (2018). *Lenguaje de programación*. [En línea] [https://es.wikipedia.org/wiki/Lenguaje\\_de\\_programaci%C3%B3n](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [Consultado el 05/12/2018]

Wikipedia. (2018). *Sistema de Leitner*. [En línea] [https://es.wikipedia.org/wiki/Sistema\\_de\\_Leitner](https://es.wikipedia.org/wiki/Sistema_de_Leitner) [Consultado el 02/11/2018]

Wikipedia. (2018). *Software* [En línea] <https://es.wikipedia.org/wiki/Software> [Consultado el 05/12/2018]

Wikipedia, (2017). *Texto bidireccional*. [En línea] [https://es.wikipedia.org/wiki/Texto\\_bidireccional](https://es.wikipedia.org/wiki/Texto_bidireccional) [Consultado el 01/11/2018]

Windows (2019). *Search Terminology*. [En línea] <https://www.microsoft.com/en-us/language/Search> [Consultado el 01/11/2018]

Wordpress, (2013). *Export TMX For New Translations*. [En línea] <https://libretraduko.wordpress.com/2013/08/10/export-tmx-for-new-translations/> [Consultado el 01/11/2018]

Wordpress.org (2018). *Translating WordPress*. [En línea] <https://translate.wordpress.org/> [Consultado el 29/10/2018]

W3schools, (1999-2018). *Non-breaking space*. [En línea] [https://www.w3schools.com/html/html\\_entities.asp](https://www.w3schools.com/html/html_entities.asp) [Consultado el 01/11/2018]

Youtube (2018). *iGUnit Flashcard Trainer Tutorials*. [En línea] <https://www.youtube.com/watch?v=LmB0YnPNSbI&list=PL-0XTgwGwyHrKyS7UMFwZoPYrh8tREe99> [Consultado el 02/11/2018]